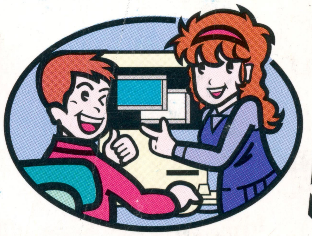
2001年11月1日発行（毎月1回1日発行）第20巻第11号 昭和57年12月6日第三種郵便物認可

マイコン

わかる！動かせる！
プログラムが組める雑誌

BASIC Magazine

NOV.2001 11



定価 550Yen

パソコン＋ロボットの新しい遊び「AIBO Nabigator for ERS-210/ERA-201D1」

AIBOを無線でコントロール！



“子どもたちに手軽にロボット工学に親しんでもらう”ために

「ロボカップトイズ」登場！

無線ブロードバンドを実践活用
群馬県前橋市立第四中学校



マンガで学ぼう！
パソコンレクチャー
トランジスタで論理回路？

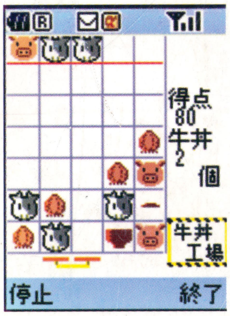
●自作一代記
①ワンダースワンにキーボードを接続
②PICプログラマーキット Ver.3

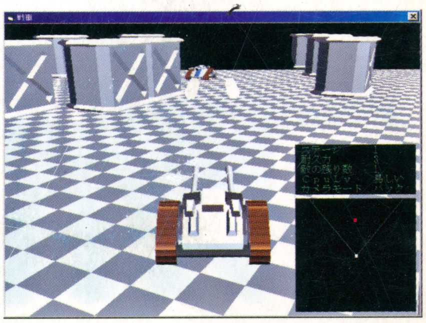
プログラマー養成講座

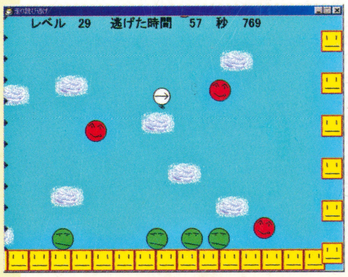
Technical News Spot
WindowsXP/インテルの新CPUテクノロジー
プログラムの疑問・質問にお答えします
PROGRAMMING Get&Run
はじめようMicrosoft ExcelでのBASICプログラミング
パズルゲームをつくってみよう
DirectXテクノロジーを伝授する！IE+DirectAnimation
ピッカプル・イベントを使って“あみだくじゲーム”を作ろう！
図解！Visual Basicゲーム作成術
キャラクター・アクション・ゲーム②方向の数値化

読者が作った作品をドーンと紹介
特選投稿プログラム コーナー

VisualBasic6.0	落下
VisualBasic6.0	Square
VisualBasic6.0	普通のタイピング・ゲーム
VisualBasic6.0	RANDOM CODE ver.1.03
VisualBasic5.0	Shino Buster BM Version
VisualBasic6.0+DirectX7	（超大作）Abandoned Ver.1.01
VisualBasic6.0+DirectX7	（超大作）戦車
HSP V2.5	走り跳び逃げ
F-BASIC6.0	四字熟語
QBasic	Translation Composer翻訳作曲家
iアプリ	ぽんこつファイト
iアプリ	牛丼工場
PC-G850S	RENDA for PC-G850S
●CMM試聴室	
MU90B	Silent
SC-8850	Fazenda









専用ホームページ
http://www.basicmagazine.net/



アンビリバボ〜
こいつぁ〜
もう、迷う
理由がないね。

マイコン BASIC Magazine

CONTENTS

11 NOV.2001

特集 パソコン＋ロボットの新しい遊び

AIBOを無線でコントロール！
「AIBO Nabigator for ERS-210/ERA-201D1」 10



マンガで学ぼう！ パソコンレクチャー　ザ・ビギニング
Step3 トランジスタで論理回路？ 16

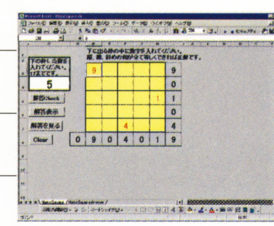
基礎編
Technical News Spot
Windows XP／インテルの新CPUテクノロジー 20

プログラマー養成講座
プログラムの疑問・質問にお答えします
PROGRAMMING Get＆Run 21

はじめようMicrosoft ExcelでのBASICプログラミング
パズルゲームをつくってみよう 24

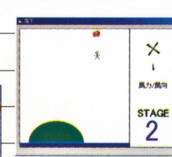
DirectXテクノロジーを伝授する！ IE＋DirectAnimationマルチメディア・プログラミング
ピッカプル・イベントを使って“あみだくじゲーム”を作ろう！ 28

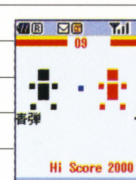
図解！ Visual Basicゲーム作成術
キャラクター・アクション・ゲーム②方向の数値化 36

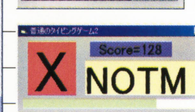


読者が作った作品をドーンと紹介 特選投稿プログラム・コーナー

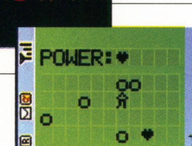
VisualBasic6.0	落下	松本輝彦	42
VisualBasic6.0	Square	岡崎　務	47
VisualBasic6.0	普通のタイピング・ゲーム	高橋好人	52
VisualBasic6.0	RANDOM CODE Ver.1.03	YUITA	54
VisualBasic5.0	Shino Buster BM Version	ていこく	59
VisualBasic6.0＋DirectX7	（超大作）Abandoned Ver.1.01	hani	63
VisualBasic6.0＋DirectX7	（超大作）戦車	橘　一馬	65
HSP V2.5	走り跳び逃げ	INF	66
F-BASIC6.0	四字熟語	伊藤　優	70
QBasic	Translation Composer 翻訳作曲家	石山明弘	72
iアプリ	ぽんこつファイト	Mr.親父	74
iアプリ	牛丼工場	Hase	80
PC-G850S	RENDA for PC-G850S	かねん	85
●CMM試聴室			87
MU90B	Silent	ナガタヒロカズ	
SC-8850	Fazenda	松田茂喜	











今月のベスト・プログラマーは誰の手に！
明日のスタープログラマー＆Programmer's Room 88

今月のベスト・プログラムを解説！
Dr.Dのベスト・プログラム研究室 90

専用ホームページ http://www.basicmagazine.net/

編集部の学校訪問記 **群馬県前橋市立第四中学校** 8

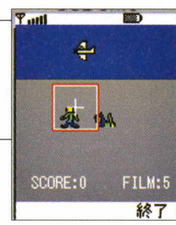
ロボ道場番外編

21世紀の遊びはこれ！
“子どもたちに手軽にロボット工学に親しんでもらう”を目的に「ロボカップトイズ」登場！ 13

好評連載　プログラミングから電子工作まで

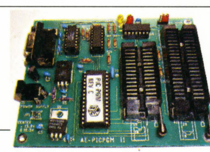
PS2用「BASIC STUDIO　パワフルゲーム工房」入門
3Dモデルを使って迷路を作る 92

Catlikeのもっと身近にプログラミング！
シューティング・ゲーム編 その3：敵艦監督を楽しもう 94



携帯電話で動くゲームを作ろう!!
キミにもできる「iアプリ」制作
タイマを使ったiアプリに挑戦 98

松原拓也の自作一代記
・ワンダースワンにキーボードを接続
・手ごろで入門者向けワンチップマイコン 102



テクノ・トイズ・ラボラトリ
今月の製作『ポケットライト・デラックス 前編 106

神奈川のりこのホームページとかに夢中　夢中48
ネット銀行に夢中 ♥ 109



レゴ道場　チャレンジ！レゴ・マインドストーム
LEGO MINDSTORMS フェスティバル 110

Linux環境でプログラミングを学ぼう!!　よいこのLinux修行
最終回：Turbolinux 7 Workstationをインストール 113

DTM Art Studio　パソコンで音楽をたのしもう！
YAMAHA「らくらく作曲名人2」～アレンジ編　その2～ 118

読者のコーナー　あくせすROOM 124

ソフト・メーカーのPRのページ
SOFTWARE MAKER HOT INFORMATION 125

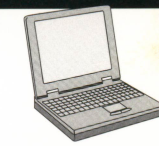
広告さくいん

ハード	
ソニー	表4
タケル	前付6
プロバイダ	
アルファインターネット	表2～前付カラー3
ソフト	
日本ファルコム	表3

学　校	
日本ゲームソフト学院	後付129
パソコンショップ	
秋月電子通商	記事中105
パシフィックネット	後付130
その他	
音楽電子事業協会	後付128

© マイコンBASICマガジン11月号
第20巻 第11号 2001年10月25日印刷
2001年11月1日発行
定価550円（本体524円）
電波新聞社　2001 Printed in Japan
編集発行人：平山哲雄
東京都品川区東五反田1-11-15
郵便番号141-8715
電話　東京(03)3445-6111（大代表）
振替　（東京）00150-3-51961
印刷：奥村印刷(株) 製本：(株)堅省堂

私たちは考えています。 パソコンのスピードアップ!!



★初心者へのパソコンの普及
これからパソコンを始める方専用のパソコン教室「ホッと倶楽部」オープンしました。

★旧型パソコンの不満解消!!　まだまだ使える旧型パソコン
ハードディスクやメモリーの性能を飛躍的に向上させ、使い慣れたパソコンを現行機種に近いかたちでお使い頂けます。

★地球環境にやさしいリサイクル
OA機器の消耗品をリサイクルすることで経費節減と産業廃棄物を削減します。

■平成13年4月27日に指示表示器付きキーボードとコンピューターシステムとして特許公開（特開2001-117704）されました。

■平成13年5月1日にフルキー表示機能付きキーボード（光るキーボード）事業は東京都・創造促進法の研究開発事業に認定（13産労商創創第2380号）されました。
※この技術を応用した事業、アイデア等にご協力頂ける企業様／SOHO様募集中!!

▼ DIME（小学館）2001/5.17号、P47の特集「あなたの中古パソコンを"活かす"3つの知得、教えます」にて紹介されました。

●ノートパソコンアップグレード（性能向上）サービス



今、お使いのノートパソコンのハードディスクを交換できることは、ご存知ですか？

少ない投資で、快適な性能・環境にアップグレードしてみませんか？

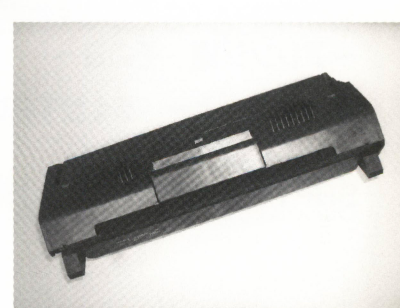
そろそろ新しい機種に変えようと思っている方、ちょっとお待ち下さい。

例えば、ハードディスクを**2GB**から**10GB**に交換することで、容量が増える他にディスクアクセス時間の高速化やデータ先読みバッファー（キャッシュメモリー）増加などの性能向上により、最新機種と変わらない性能になります。　費用は**￥29,800**より

◎容量→**5倍**、速度→**2.5倍**（内部のディスクアクセス時間）、キャッシュメモリー→**4倍**

※ その他、パソコンのアップグレードに関することお問合せ下さい。

●リサイクルトナー



レーザープリンターやレーザーFAXのトナーカートリッジを使い捨てにしていませんか？
貴社の経費節減と環境保護対策の一貫として、是非、リサイクルをご検討下さい。
基本的には空カートリッジをお預かり後、1～2週間で納品いたしますが在庫を持っておりますのでご希望であれば**即納も可能**です。
また、純正品についてもお安くお出しすることも可能です。
当社のリサイクルトナーならびに純正トナーを、是非、一度お試し下さい。

●リサイクルトナー　　メーカー標準価格の**約10％～70％OFF**

●取り扱いメーカー（国内外のほとんどのレーザープリンター／レーザーファックスはリサイクル可能）
CANON、EPSON、FUJITSU、NEC、OKI、XEROX、RICOH、APTi、QMSなど

※事業拡大により人材募集中！
ソフトウェア、ハードウェア技術者、ASIC開発技術者、その他アルバイトも同時募集

お問合せは

Takeru Co.,ltd

http://www.takeru-net.co.jp

株式会社タケル

〒110-0016
東京都台東区台東2-7-12　ミツワビル4F
TEL　03-5817-0675
FAX　03-5817-0977
E-Mail　office@takeru-net.co.jp



▲3年生選択技術でロボット製作に取り組むみなさん

群馬県 編集部の学校訪問記
前橋市立第四中学校

http://www.daiyon-jhs.menet.ed.jp/

無線ブロードバンドを実践活用！
最近はロボット製作に取り組む生徒たち



▲情報教育には欠かせない技術家庭科担当の小柏先生



前橋市は全学校が無線ネットワークで結ばれています

「前橋市教育情報ネットワーク」（MENET）無線ネットワーク構成図
MENETホームページ　http://www.menet.ed.jp

「先進的教育用ネットワークモデル地域事業」のモデル地域・前橋市

群馬県前橋市は，平成10年度から実施されている文部省（現・文部科学省）と郵政省（現・総務省）の連携事業「先進的教育用ネットワークモデル地域事業」のモデル地域に指定され，市内各学校・教育機関を高速アクセス回線で結ぶ「前橋市教育情報ネットワーク」（MENET）を構築しています。

前橋市立第四中学校は平成7年にインターネット工事を実施した，この事業の先駆け的存在。各教科でインターネットを使った情報収集を採り入れ，メールの使いかたやホームページの作成など積極的に活動しています。

「夢は大きく　志は高く　やればできる」をスローガンに，318人の生徒がのびのびと勉強に運動に励んでいます。

MENETはWLL（ワイヤレスローカルループ）と呼ばれるパラボラアンテナなどを使い，市内60数か所の小・中・高校，大学，施設などを無線ネットワークで結んでいます。前橋市総合教育プラザ内のサーバで各学校のデータを管理し，接続方法が22GHz無線（WLL）の場合は通信速度1.5Mbps，2.4GHz無線ルータの場合は通信速度2～11Mbpsという環境です。そのため，指定校は快適な通信環境のなかでコンピュータ学習が可能になっています。



▶コンピュータ室には昨年入れ替えられたエプソンのタイプSL2が40台，LAN接続されている



◀第四中学校のホームページ



▲小柏先生の指導のもと，ロボット製作に取り組む生徒たち



▲知恵を絞りながら作るのでコミュニケーション能力も養われる



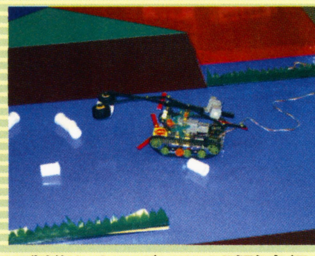
▲フレーム部分が完成！



▲シャベルの部分は慎重に



▲自慢のマシンを微調整



▲製作したロボットの試験走行



▲大半がブルドーザータイプなのに対し，はさむタイプは珍しい



▲小柏先生と第2回校内ロボコン大会の打ち合わせ中

1.5Mbpsの無線インターネットで快適学習

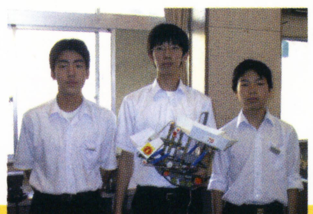
当然，先駆者である前橋第四中学校のコンピュータ室には，パソコンが40台設置され，すべてがLAN接続されています。同校の通信速度は1.5Mbpsです。「2年生と3年生全員にメールアドレスが与えられ，メール交換やホームページの作成，調べ学習などに活用しています」とは技術・家庭科と情報教育を担当する小柏（おがしわ）哲也先生。

同校では技術の授業でパソコンの基礎（ワープロやインターネット，メールなど）を学び，高学年になるとパソコンを利用した研究発表や修学旅行の報告，思い出アルバムの制作などを行います。それぞれが作成したホームページなどは校内サーバで公開されます。

また，科学部の研究発表や美術部での作品作り，3年生の卒業記念に作るTシャツのデザインなどもパソコンは欠かせない存在になっているようです。

ロボコン優勝チームも登場！選択技術で取り組むロボット製作

取材時は，3年生の選択技術の授業を拝見しました。小柏先生は「選択技術は普段の技術の授業ではできない機械領域の内容を自主的に深く掘りさげて勉強することを目標にしている」と言います。昨年からは2年生と3年生は，自主的にロボットの制作に取り組み，前橋市のロボットコンテスト（ロボコン）や県主催のロボコンにもチャレンジしています。そして今年，前橋市の第2回大会「まえばしロボコン2001」では，念願の優勝チームも誕生しました。中学生の部113チーム参加の中での堂々の1位です。



▲まえばしロボコン2001で優勝したチーム「ぐるぐるTHK」のみなさん（中央がロボットの製作をした安達拓也君，左が小島英之君，右が原澤広多君）



▶昨年度の校内大会のもようがホームページで紹介されている

授業では2～3人が1チームになって，それぞれが試行錯誤しながらロボット製作をしています。「インターネットでわからないことを調べたり，情報を収集して，それぞれのチームでオリジナルのロボットを製作。最後に製作記録や大会の結果などをホームページで公開するという流れで学習しています」と小柏先生。

今年のロボコンは，製作したロボットでペットボトルや空き缶，フィルムケースなどのゴミを分別ボックスに入れるというルールで，製作されたロボットはブルドーザーのように前方にシャベルが付いたものやゴミをはさむタイプなどさまざまでした。

ロボットは基本的に何を使用して製作してもよいとのことですが，山崎教育システムの「遊星スピードユニット」（タイヤ型）と「遊星パワーユニット」（キャタピラ型）といった製作キットを基に改造していくケースが多いようです。安定した制動力，フィルムケースやペットボトルを分別ボックスまで落とさずに運ぶための荷台などなど，みなさんのアイデアが光っていました。

校内ロボコン大会を開催するなど，生徒の自主性がきわ立つ

授業で作ったロボットは市や県主催のロボコンに出場するだけでなく，校内のロボコン大会でも活躍します。まもなく第2回校内ロボコン大会が開催される予定で，現在ルール作りなどを決めている最中でした。

また，11月には校内文化発表会を開催する予定で，そこでのデモンストレーションにロボットが活躍します。みなさん，こうした催しに向けロボットの調整や仕様変更などに一生懸命でした。

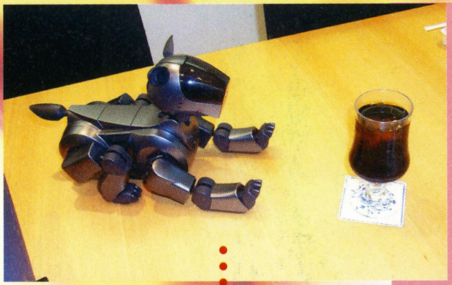
前橋市立第四中学校 〒371-0017 前橋市日吉町3-9-2 電話027-231-6743



特別企画 パソコン+ロボットの新しい遊び AIBOを無線でコントロール！

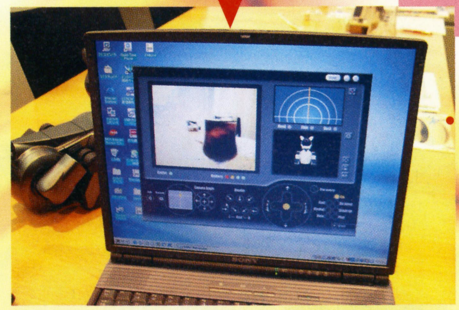
（アイボ・ナビゲータ）

「AIBO Navigator for ERS-210/ERA-201D1」



「AIBO Navigator」はパソコン上からAIBOをコントロールすることができるAIBO-wareです。

この「AIBO Navigator」は2001年3月号で紹介した「AIBOマスタースタジオ」とちがって，プログラミングの知識を必要とせず，マウスやジョイスティックで直接動かすことができます。



▲AIBOの見ている映像がパソコンにリアルタイムで表示される!!



▶パソコン上の操作画面。左上が「カメラ実況エリア」

AIBOの視点を堪能！

コントロールは無線LANを使って行うため，AIBOのPCカードスロットに「ワイヤレス LANカード」を装着させます。無線LANは「IEEE802.11b」を採用しています。このため，かなり広範囲にAIBOと通信することができます。

AIBOに内蔵されているカメラが写した映像（AIBOの視点）は，パソコンソフト上の「カメラ実況エリア」というウインドウに表示されます。映像は少々粗いですが，リアルタイムに更新されるので見やすいです。

歩行以外も充実

コントロールできる要素は充実しています。まず，前進，バック，右回転，左回転などの移動。移動速度は3段階に調



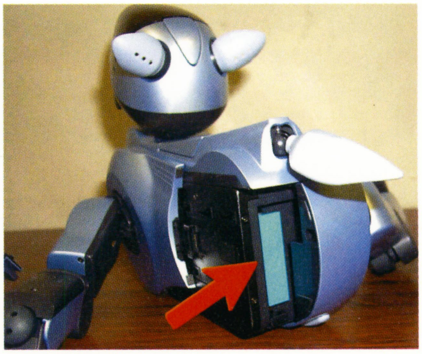
▲推奨しているジョイスティックはマイクロソフト製の「サイドワインダーフォースフィードバック2」。トリガやスロットルがAIBOの各動きに割り当てられている

「AIBO Navigator for ERS-210/ERA-201D1」



ERF-210AW05
価格：15,000円（税別）
パッケージ内容：AIBO-ware "メモリースティック"，アプリケーションCD-ROMなど
対応OS：Windows ME/98SE/2000
CPU：Intel Celeron 300MHz相当以上。
無線LAN（IEEE802.11b準拠）対応のPC，または無線LANのアクセスポイントとLAN接続されたPCが必要。

※製品にジョイスティックは付属しません。
※AIBOにはAIBOワイヤレスLANカード ERA-201D1が必要です。
※ERS-210（2nd GENERATION）専用です。ERS-310（ラッテ／マカロン）シリーズには対応していません。



▲矢印の部分が「ワイヤレス LANカード ERA-201D1」。価格は20,000円（税別）



◀スタンドの上でも愛想をふりまくAIBO達（WORLD PC EXPO 2001より）

整できます。そして，移動とは独立して首の向きだけを動かせます。その他，キック，ヘディング，お座り，ダンスやパフォーマンスなどをさせることができます。音転送機能でAIBOに声でしゃべらせたり，お気に入りのカメラ画像を保存したりすることが可能です。

コントロールに挑戦

実際に「AIBO Navigator」を使ってAIBOを動かしてみました。操作はマウス以外にもジョイスティックに対応しています。ジョイスティックのレバーを前に倒し続けるとAIBOが歩きはじめます。横に倒すと方向転換。トリガを押すとキックやヘディングをします（キーアサイン※は変更可能）。

複雑な操作をジョイスティック1本でほとんど操作することができますので，さながらラジコン戦車で遊んでいる感じです。

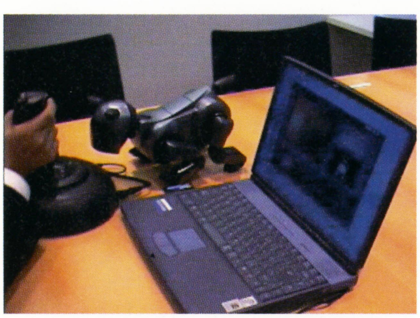
それと，実験として，AIBOを遠くに置いて，パソコン上の映像だけを見ながらコントロールしてみました。歩きながらだと映像がかなり揺れるのがつらいですが，探索している気分を十分に味わうことができます。

この他，ソフトにはピンクボールの検出機能がありますので，AIBO同士のサッカー対戦に応用してみるとおもしろそうです。

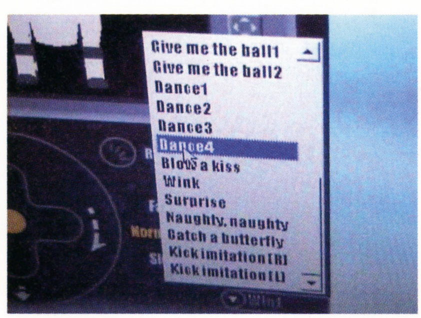
※キーアサイン＝どのキーによって，どんな機能が起動されるかという対応関係



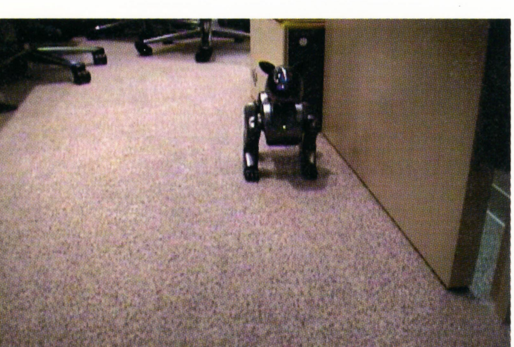
▲ジョイスティックの操作で歩きだすAIBO！



▲ヘディングやキックはトリガを押すだけで実行される



▲数多くのダンスやパフォーマンスが登録されている



▲無線LANの通信可能距離は屋内で約20m（障害物のまったくないところでは100m）。

使い道募集！

この「AIBO Navigator」によって，画期的な技術がAIBOにもたらされたわけですが，使い道がとくに決まっているわけではなく，ユーザーが自由に発見するというスタンスになっています。

そこで，突然ですが「AIBO Navigator」の楽しい使いかたを募集したいと思います。本誌とじ込みの愛読者ハガキか，官製ハガキに「こんなことに使えそう」というアイデアを書いて（現実的・非現実的を問わず）ベーマガ編集部までお送りください。

優秀作品はソニーさんに提案してみますのでよろしく。

AIBO「ERS-310」シリーズ

ラッテ「ERS-311」／マカロン「ERS-312」登場!



▲AIBOの新しい仲間，（左側）「マカロン（ERS-312）」と（右側）「ラッテ（ERS-311）」

エンターテインメントロボット「AIBO」に新しい仲間「ERS-310」シリーズが加わりました。色はアイボリーとグレーの2種類で，名前は「LATTE（ラッテ）」と「MACARON（マカロン）」です。

注目はデザイン

今度のAIBOは頭が丸く，クマイヌ型のデザインです。頭上の「角ランプ」は発光し，色と点滅の種類で感情や状態を表現します。

音声認識機能の強化

音声認識できる言葉が約75種類（今までの約1.5倍）に増えました。他にもオーナー（飼い主）の名前を覚えさせたり，AIBO「ERS-311/312」同士でAIBO語によるコミュニケーションができる機能が追加されています。

その他の機能

鼻先に手をかざすと，その距離に応じて音を生成。また，「ラララ」と歌いかけると，その音階をもとにAIBOが歌って返すことがあります。

それと，AIBOが写真を撮影してくれる写真撮影機能もあります。写真はメモリースティックに保存されます（JPEG形式）。

2種類のAIBO-ware

AIBO「ERS-310」シリーズには「AIBOフレンド」と「AIBOライフ」という2種類のAIBO-ware“メモリースティック”があります。

AIBOフレンド「ERF-310AW06J」はラッテとマカロンで別々の性格付けがされています。ラッテは素直でおっとりした性格，マカロンは陽気でやんちゃな性格です（性格はその後も変化します）。

その他，「ラブモード」「メディアリンクモード」「おるすばんモード（別売のエナジーステーションコアが必要です）」など，盛りだくさんな機能があります。

AIBOライフ「ERF-310AW01J」は幼児から成年までの成長ステージを楽しむことができるプログラムです。成長する段階ごとに，いろいろな性格のAIBOに変化していきます。

AIBOと遊んでみる

実際に遊んでみました（AIBOフレンドを使用）。音声認識で「お手」「ご挨拶」「ラララ」などの言葉に愛嬌のある反応をします。

価格は98,000円，AIBOがより親しみやすくなりました。

■AIBO「ERS-311/ERS-312」

■価格 98,000円（税別）

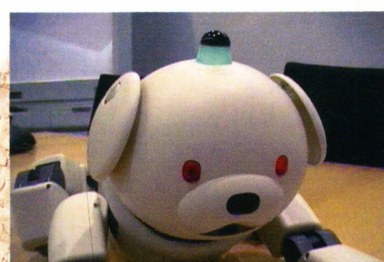
■性能表

重量	約1.5kg（バッテリー・"メモリースティック"含む）
寸法（幅・高さ・奥行き）	約177×280×240mm
CPU	64bit RISCプロセッサー
主記憶	32MB
プログラム供給媒体	AIBO専用"メモリースティック"
可動部	頭：3自由度，脚部：3自由度×4脚，計15自由度
入力スイッチ	音量調節スイッチ
画像入力	10万画素CMOSイメージセンサー
音声入力	ステレオマイクロホン
音声出力	スピーカー
内蔵センサー	赤外線方式測距センサー，加速度センサー，スイッチ（頭部内，しっぽ，肉球），振動センサー，傾斜センサー

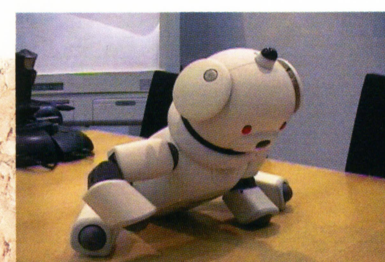
■性格変化 AIBO-ware“AIBOフレンド（ERS-311/312用）”「ERF-310AW06J」価格 9,000円（税別）

■自律成長 AIBO-ware“AIBOライフ（ERS-311/312用）”「ERF-310AW01J」価格 9,000円（税別）

■AIBO HomePage
http://www.aibo.com/



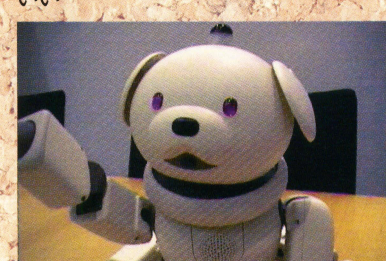
▲頭上の角ランプ（モードランプも内蔵），鼻には距離センサ，口にカラーカメラを搭載。両耳のステレオマイクは音の方向を検出可能



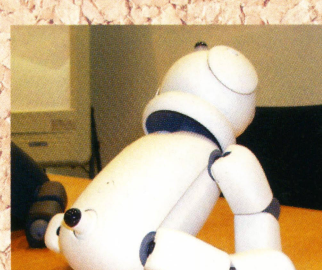
▲ご挨拶するAIBO。動きが非常にかわいい



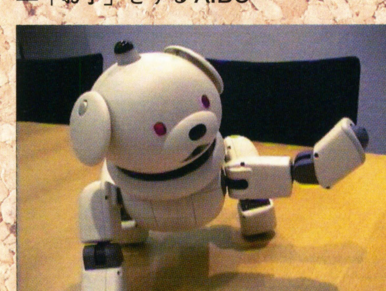
▶足の裏にある「肉球スイッチ」



▲「お手」をするAIBO



◀おしりには「しっぽスイッチ」，背中には「背中ランプ」「ポーズ（一時停止）ボタン」を装備



▲バランスを保ち，3本足で立つAIBO

TV新番組「ピロッポ」 メディアリンク機能を取り入れたアニメーション番組です。AIBOも登場するそうなのでお楽しみに。
放送日：第1回目10/11（木）23:09～23:15，2回目以降の放映は毎週木曜日22:54～23:00　放送局：フジテレビ系列
※メディアリンク機能とは，テレビやインターネットなどからの特定のメロディ信号にAIBOが反応する機能です。

21世紀の遊びはコレ！ ロボ道場 番外編

“次世代を担う子供達に手軽にロボット工学に親しんでもらおう！”を目的に！

タカラ，トミー，バンダイ3社から「ロボカップトイズ」登場！



今月はいよいよ発売された「ロボカップトイズ」を特集します。　（松原拓也）

ロボカップトイズ（RoboCup Toys）

タカラ，トミー，バンダイの3社が「ロボカップ」（ロボット工学と人工知能の融合・発展のために自律移動ロボットによるサッカーを題材として，日本の研究者らによって提唱された）の理念に賛同して「ロボカップトイズ」を創設。各社から一斉にマシンが発売されました。

マシンの種類は現在3タイプ。見た目はラジコンカーみたいですが，立派なサッカーロボットです。赤外線式のコントローラでマシンを操り，サッカー競技を行います。

公式ルールではフィールドのサイズは81×108cmと108×274cmの2種類。対戦人数は1vs1，2vs2，3vs3の3種類です（性能的には4vs4も可能）。ロボットのサイズは直径33cm×高さ30cmまでです。



タカラ
「ジェイストライカー」
寸法：240×150×82mm
色：ブルー／ホワイト
価格：4,980円（税別）



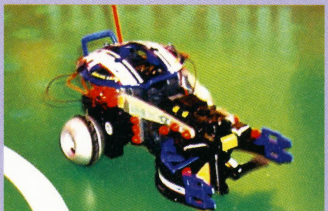
トミー
「ブレイクリーガー」
寸法：275×152×112mm
色：スカイブルー／ファイアーレッド
価格：4,980円（税別）



バンダイ
「フィールドコマンダー」
寸法：275×156×100mm
色：ブラック＆レッド／ホワイト＆ブルー
価格：4,980円（税別）



▲上体を持ち上げる「リフト機構」を搭載



▲ボールのホールド機能がなく，速攻性に優れたマシン



▲ツメでボールを挟みこむ「クロスホーンシューター」を搭載

マシンの組み立て



セットにはマシンとコントローラ，ケーブル，そしてボールが付属します(電池は別売)。最初，マシンはバラバラの状態ですので，説明書を見ながら組み立てます。

◀購入時はパーツ単位でバラバラ。説明書を見ながら組み立てます

マシンは「コアユニット」「バッテリーボックス」「メインフレーム」「ボディ」「走行ユニット」「シュートユニット」というパーツに分かれていて，それらを「ロックピン」でつなぎ合わせます（接着剤不要で便利)。規格が統一されていますので，パーツを他のマシンから流用することもできます。組み立ては簡単で15分くらいで完成しました。

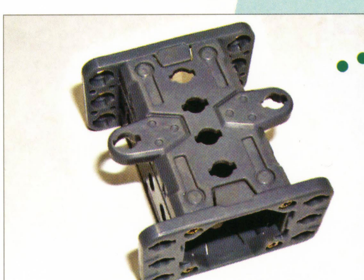


▲「コアユニット」(全社共通)。赤外線の受光部とマイコンを内蔵しています。中に9V電池1個を装てんします

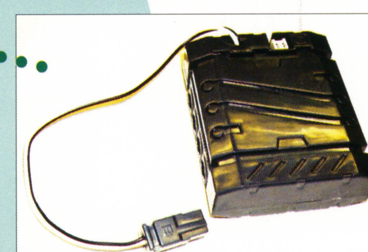




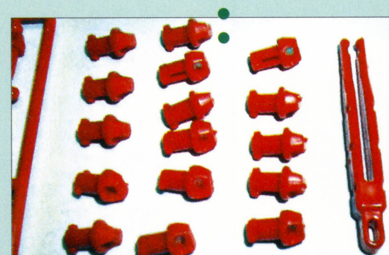
▲「ボディ」。外装です。付属のシールを貼ってカッコよくしましょう



▲「メインフレーム」(全社共通)。各パーツの中心になる骨組みです



▲「バッテリーボックス」(全社共通)。駆動用の電池ボックスです。単3電池3本を装てんします



▲「ロックピン」。パーツの留め具です。ジョイントツールを使って90度回転させると固定されます



▲「シュートユニット」。ボールの発射装置です。モータを内蔵しています

注）この記事はバンダイ「フィールドコマンダー」で製作しています。パーツ名は3社とも共通ですが，形状がちがいます。ただし，写真説明文に(全社共通)とあるのは，形状も同じことを意味します。



▲「走行ユニット」ギアを内蔵した移動用のモータです。ユニットは右用と左用で計二つあります

遊んでみる

マシンは付属のコントローラで操作します。方向キーを押すと,「ギュイーン」と音を立ててマシンが移動します。Aボタンでボールをつかみ，発射させることができます(60cmくらい飛びます)。Bボタンでアームを持ち上げれば，ループ・シュートも可能です(「フィールドコマンダー」の場合)。

マシンはバンドを調整することで最大8台まで操作できます。



▲付属のコントローラ。操作は方向キーとA/Bボタン（C/Dボタンは拡張用)。操作可能距離は約3m



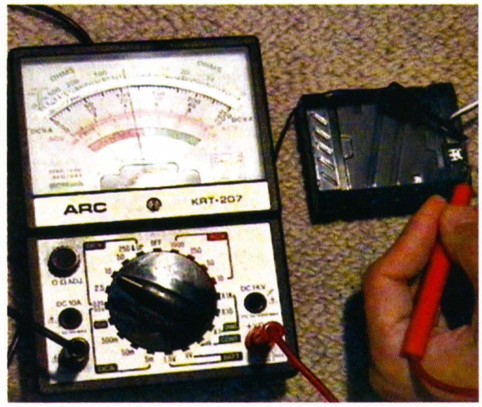
▲同時に操作するときは，コントローラ同士をケーブルで接続する（最大八つまで)

オリジナル・ユニットの製作

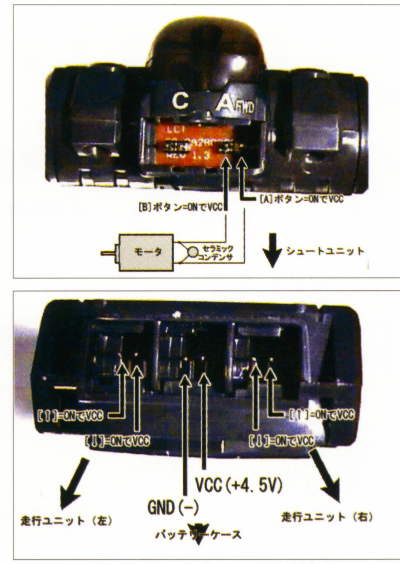
（※改造する場合は自己責任で注意して行ってください）

ロボカップトイズではルール内であれば自作パーツが認められています。まだ具体的な製作方法がアナウンスされていませんが，ちょっと先走ってオリジナルのシュートユニットを作ってみました。

まずは，テスターで電圧を測定してみました。バッテリユニットから出ている4.5Vは，コアユニットのモータプラグに通じています。これがオンかオフとなってシュートユニットや走行ユニットを動かしているようです。



▲テスターの使いかた。「電圧計」にレンジを合わせ，－端子をグランドに，＋端子を測定部分に当てます



▲コアユニットのピン配置（独自に調べたものです）

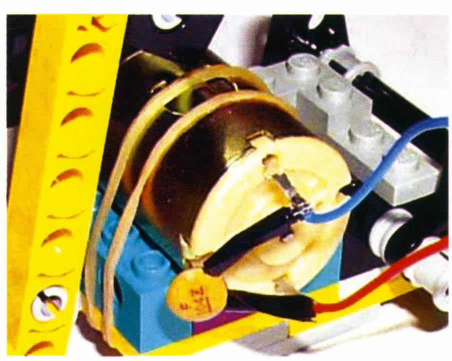
ロボ道場 マブチモータ版

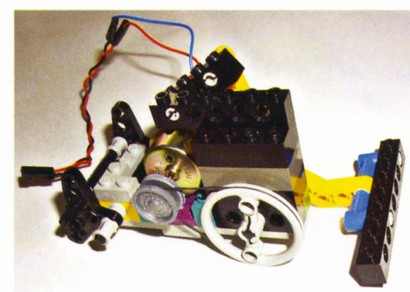


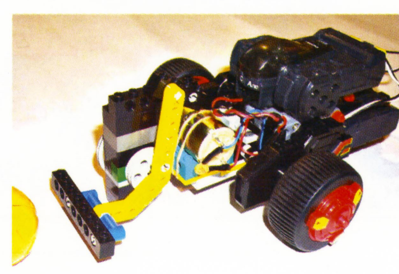
▲マブチモータ製「RE-280」。動作電圧は1.5〜4.5V。東急ハンズで200円（税別）で購入

4.5Vの電圧に耐えられるモータとして，マブチモータ製の「RE-280」を買ってきました。モータの両端には0.1μF程度のセラミックコンデンサ（約10円）をはんだ付けします。モータプラグは2.54mmピッチのコネクタを流用しました。ボールをキックするメカ部分はおなじみのレゴ・マインドストームで作ってみました。オリジナルシュートユニットの完成です。

A/Bボタンを押すと，モータが回転します。この「RE-280」では消費電流が大きく，回転に勢いがつきすぎのようです。


▲モータの保護用にセラミックコンデンサを付ける

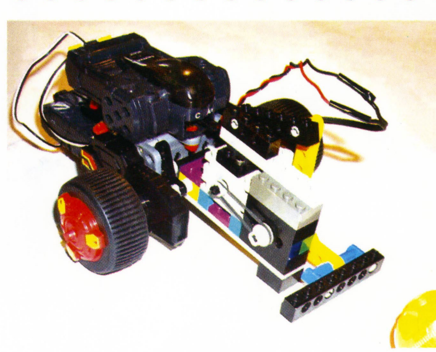

▲モータの回転をゴムベルトで伝え，クランク運動でアームを動かす

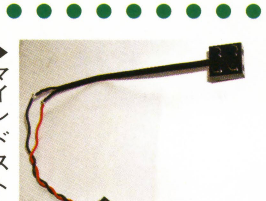

▲オリジナルシュートユニットの完成。A/Bボタンを押すと，激しくアームが動く!!

ロボ道場 レゴ製9Vモータ版

そこで今度はマインドストーム用の9Vモータをつないでみました。9Vモータのコネクタはレゴ製ケーブルを切断して使いました。パーツが統一されて，見た目が美しいですが，回転が弱いので，プーリーの比で調整しました。

この要領でレゴブロック製の走行ユニットも作れますが，試合に使うとぶつかってバラバラになりそうです。


▲こちらはマインドストーム用9Vモータを搭載したタイプ。駆動電圧が4.5Vなので回転が遅め


▶マインドストーム用ケーブルを切断して自作したモータプラグ（接合部分は絶縁しよう）

最後に

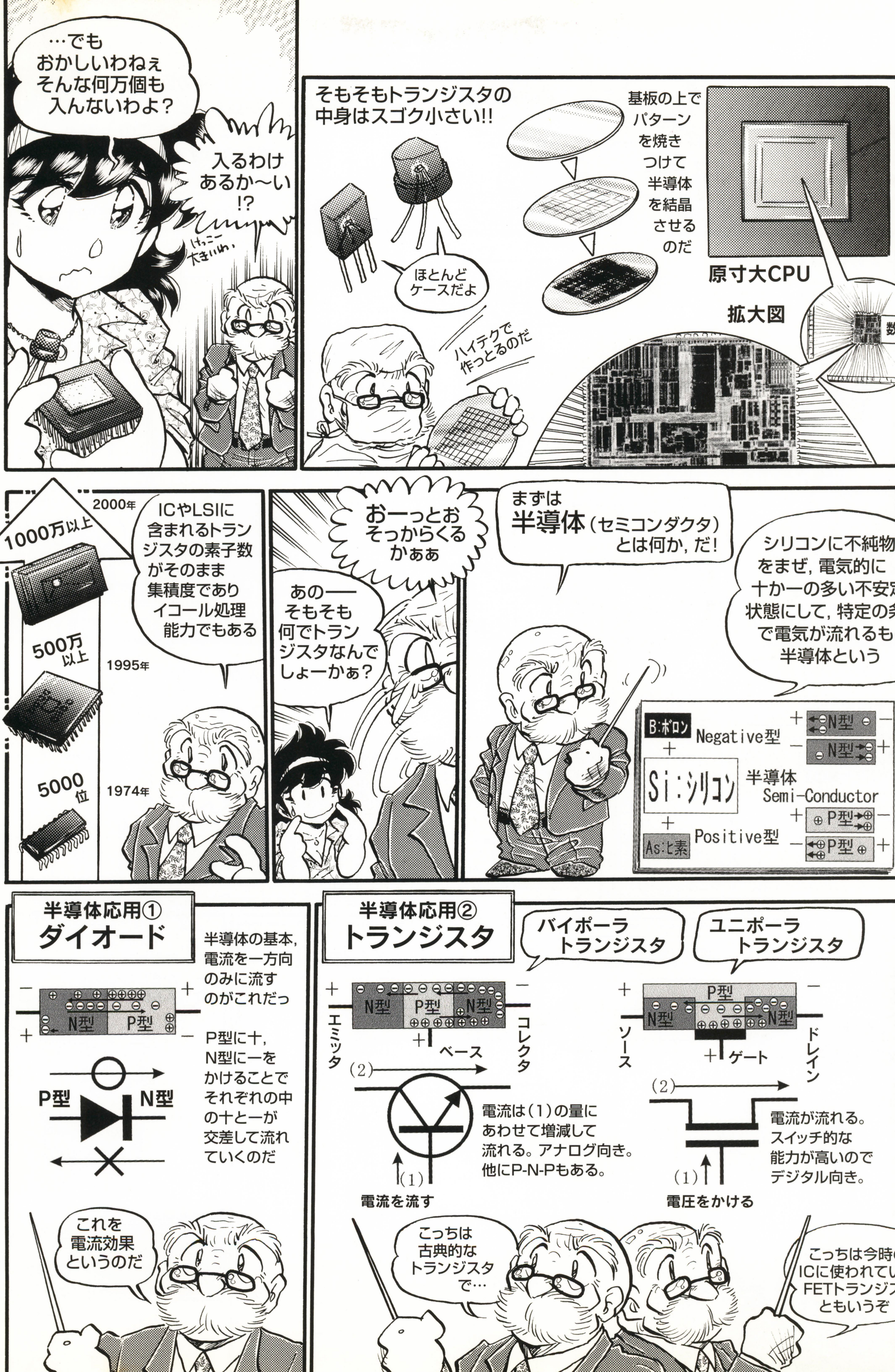
多人数で遊べ，カスタマイズがいろいろできる点がおもしろいですね。2002年には全国大会が行われますので，オリジナル・マシンで最強を目指してみてはいかがでしょうか。

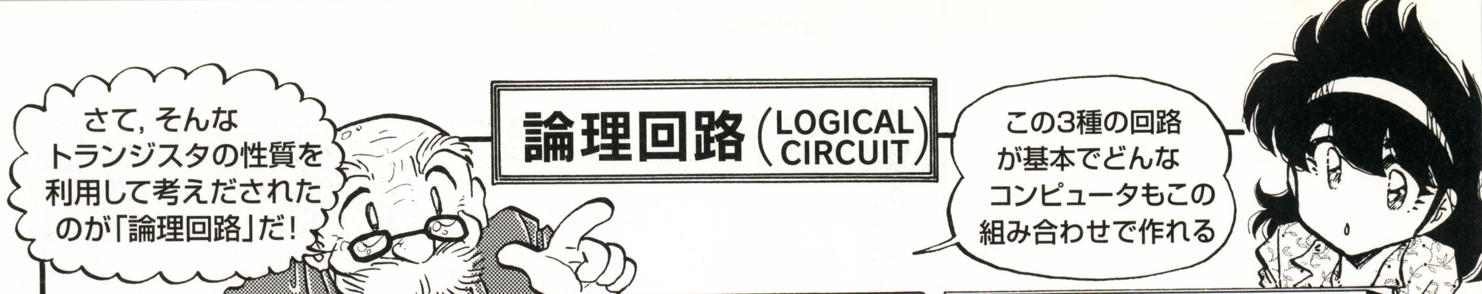
パソコンレクチャー ザ・ビギニング

Step3：「トランジスタで論理回路？」

by くりひろし

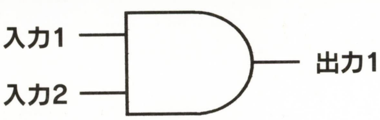


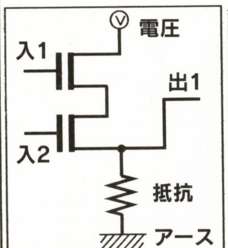

…でも おかしいわねぇ そんな何万個も 入んないわよ？
入るわけ あるか〜い !?
けっこー 大きいわ、
そもそもトランジスタの 中身はスゴク小さい!!
ほとんど ケースだよ
基板の上で パターン を焼き つけて 半導体 を結晶 させる のだ
ハイテクで 作っとるのだ
原寸大CPU
拡大図
数ミリ
2000年
1000万以上
500万 以上
1995年
5000 位
1974年
ICやLSIに 含まれるトラン ジスタの素子数 がそのまま 集積度であり イコール処理 能力でもある
おーっとお そっからくる かぁぁ
あのー そもそも 何でトラン ジスタなんで しょーかぁ？
まずは 半導体（セミコンダクタ） とは何か，だ!
シリコンに不純物 をまぜ，電気的に 十か一の多い不安定な 状態にして，特定の条件 で電気が流れるものを 半導体という
B:ボロン
Negative型
Si:シリコン
半導体 Semi-Conductor
As:ヒ素
Positive型
N型
P型
半導体応用① ダイオード
半導体の基本， 電流を一方向 のみに流す のがこれだっ
P型に十， N型に一を かけることで それぞれの中 の十と一が 交差して流れ ていくのだ
N型 P型
これを 電流効果 というのだ
半導体応用② トランジスタ
バイポーラ トランジスタ
ユニポーラ トランジスタ
N型 P型 N型
エミッタ
ベース
コレクタ
(2)
(1)
電流を流す
電流は（1）の量に あわせて増減して 流れる。アナログ向き。 他にP-N-Pもある。
P型
N型 N型
ソース
ゲート
ドレイン
電圧をかける
電流が流れる。 スイッチ的な 能力が高いので デジタル向き。
こっちは 古典的な トランジスタ で…
こっちは今時の ICに使われている FETトランジスタ ともいうぞ



論理回路（LOGICAL CIRCUIT）

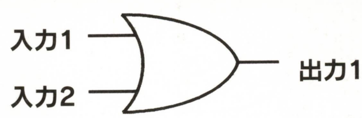
AND（アンド：論理積）

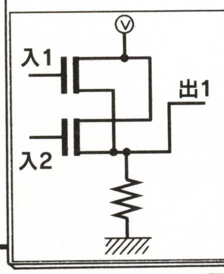




入力1	入力2	出力1
0	0	0
0	1	0
1	0	0
1	1	1

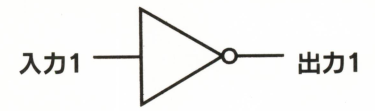
OR（オア：論理和）

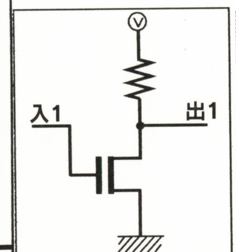




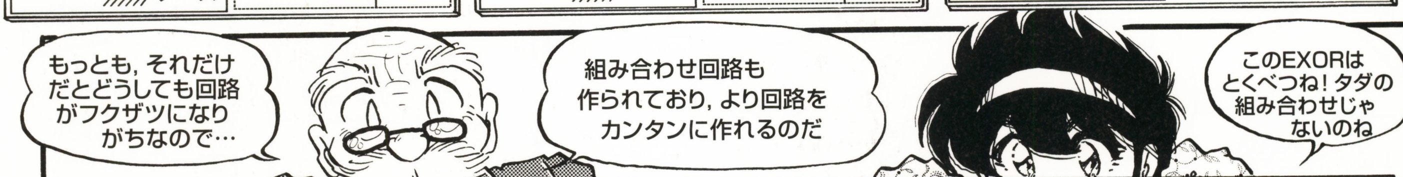
入力1	入力2	出力1
0	0	0
0	1	1
1	0	1
1	1	1

NOT（ノット：否定）

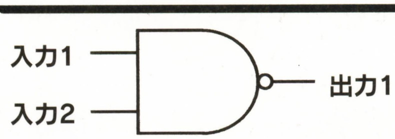




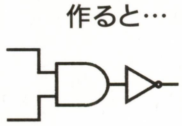
入力1	出力1
0	1
1	0



NAND（ナンド）

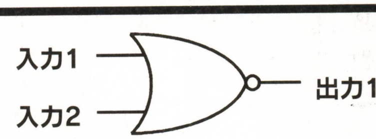


NANDをANDとNOTで作ると…

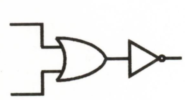


入力1	入力2	出力1
0	0	1
0	1	1
1	0	1
1	1	0

NOR（ノア）

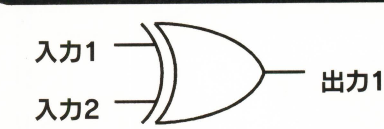


NORをORとNOTで作ると…

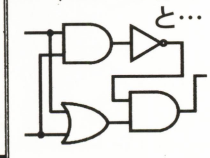


入力1	入力2	出力1
0	0	1
0	1	0
1	0	0
1	1	0

EXOR（イクスクルーシブオア：排他的論理和）



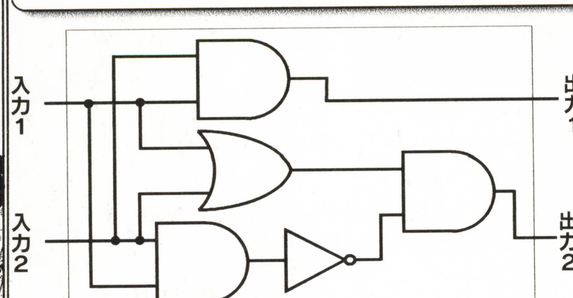
EXORをAND，OR，NOTで作ると…



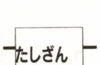
入力1	入力2	出力1
0	0	0
0	1	1
1	0	1
1	1	0

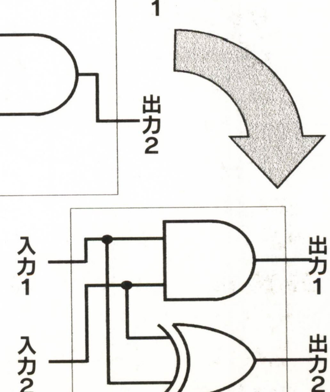


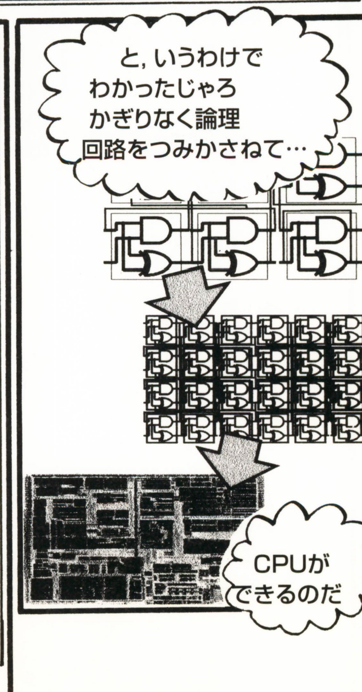
足し算回路（半加算器）を論理回路で作ると…

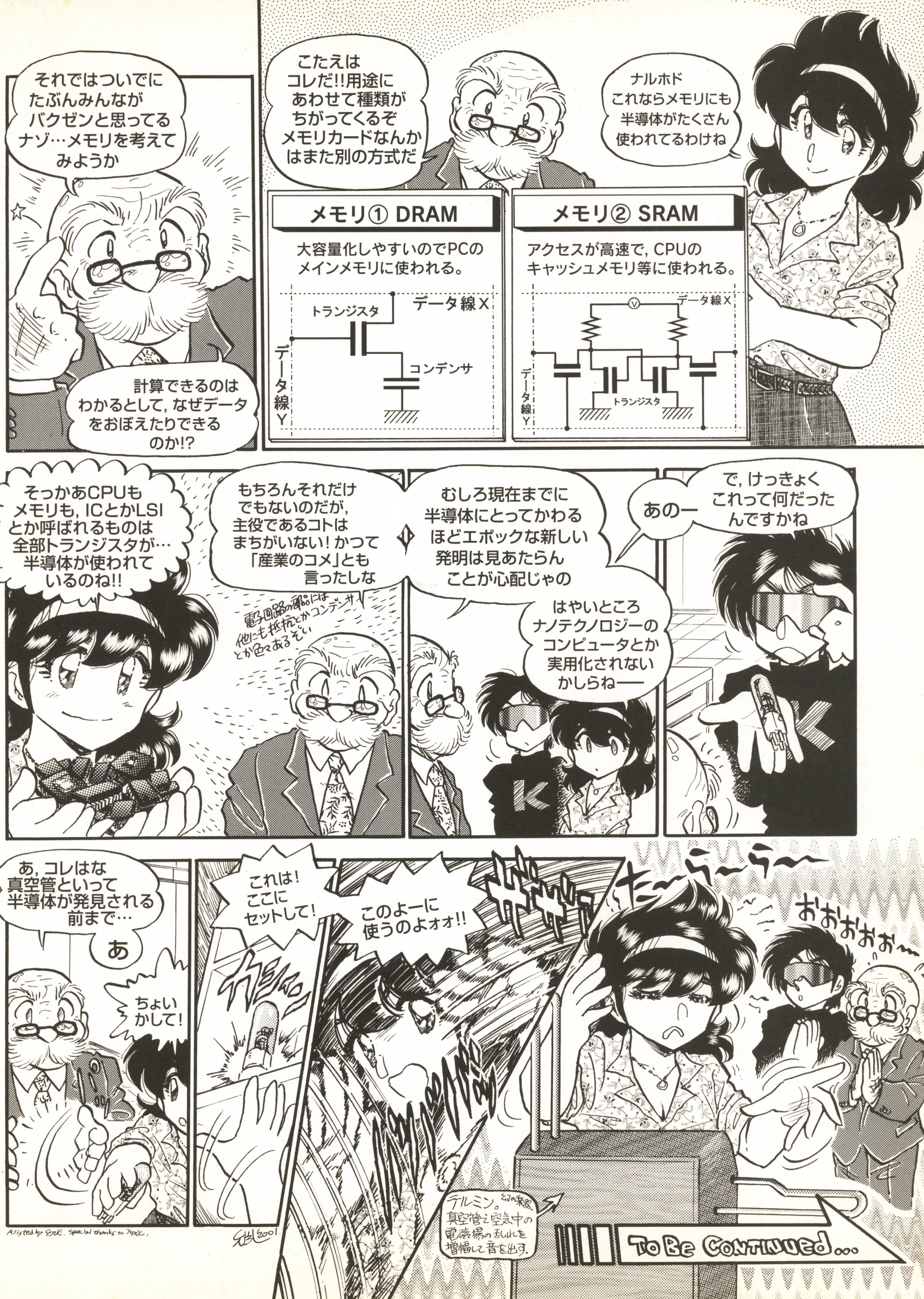


入力1	入力2	出力1	出力2
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0








それではついでにたぶんみんながバクゼンと思ってるナゾ…メモリを考えてみようか
計算できるのはわかるとして, なぜデータをおぼえたりできるのか!?
こたえはコレだ!!用途にあわせて種類がちがってくるぞメモリカードなんかはまた別の方式だ
ナルホドこれならメモリにも半導体がたくさん使われてるわけね
メモリ① DRAM
大容量化しやすいのでPCのメインメモリに使われる。
トランジスタ
データ線X
データ線Y
コンデンサ
メモリ② SRAM
アクセスが高速で, CPUのキャッシュメモリ等に使われる。
データ線X
データ線Y
トランジスタ
そっかあCPUもメモリも, ICとかLSIとか呼ばれるものは全部トランジスタが…半導体が使われているのね!!
もちろんそれだけでもないのだが, 主役であるコトはまちがいない! かつて「産業のコメ」とも言ったしな
電子回路の部品には他にも抵抗とかコンデンサとか色々あるぞい
むしろ現在までに半導体にとってかわるほどエポックな新しい発明は見あたらんことが心配じゃの
はやいところナノテクノロジーのコンピュータとか実用化されないかしらねー
あのー
で, けっきょくこれって何だったんですかね
あ, コレはな真空管といって半導体が発見される前まで…
あ
ちょいかして!
これは! ここにセットして!
このよーに使うのよォ!!
た〜ら〜ら〜
おおおおお〜
テルミン。幻の楽器。真空管と空気中の電磁場の乱れを増幅して音を出す。
To Be Continued...
2001.

マイクロソフトのWindows XP／インテルの新テクノロジーと年末は楽しみだ！

テクスポ　2001年11月号

ついにWindows XPの日本語版が完成しました。そして，インテルは技術発表会において，いくつかの新技術を発表しています。IT不況といわれる中でも，大手企業は元気なようですね。今回は，もう一つ最近流行しているウイルスについてのお話もあります。

TEXT●石川直人



Delphi使い。埼玉に引っ越すことになりました。引っ越し準備で，忙しい毎日です。体重は，久しぶりの草野球で完投したら，一時的に73.2kgまで減りました（すぐにもとにもどりましたが…）。

新製品

Windows XP日本語版が完成！

マイクロソフトが開発していたWindowsの次期バージョン『Windows XP日本語版』が，ついに完成しました。すでにメーカーには出荷が開始されており，一般販売の開始は11月16日となっています。バージョンは二つあり，企業向けのProfessional（プロフェッショナル）版（価格35,800円），個人向けのHome Edition（ホーム エディション）版（価格25,800円）が用意されます（価格は店頭予想）。

発売日の深夜には，また，秋葉原などに行列ができるのでしょうか？

解説

これがインテルの新CPUテクノロジーだ！

低価格化競争が激しいCPU業界ですが，インテルは新しい技術の開発にも積極的です。

インテルの新しいCPUでは，以下のような技術によって性能を強化するようです。

- CPUの中で，使っていない部分の動作クロックを遅くすることで省電力化する
- CPU命令の組み合わせを工夫し，同時実行時のパフォーマンスを高める
- CPU内部でスレッド単位の並列処理をサポートする

とくに，3番目の技術は非常に先進的なものです。もう少しくわしく説明すると，**第1図**のようになります。

これら技術が搭載されたCPUが発売されるのはまだ先の話ですが，今から登場が楽しみな技術ですね。

ニュース

ブロードバンド時代はウイルスが大流行？

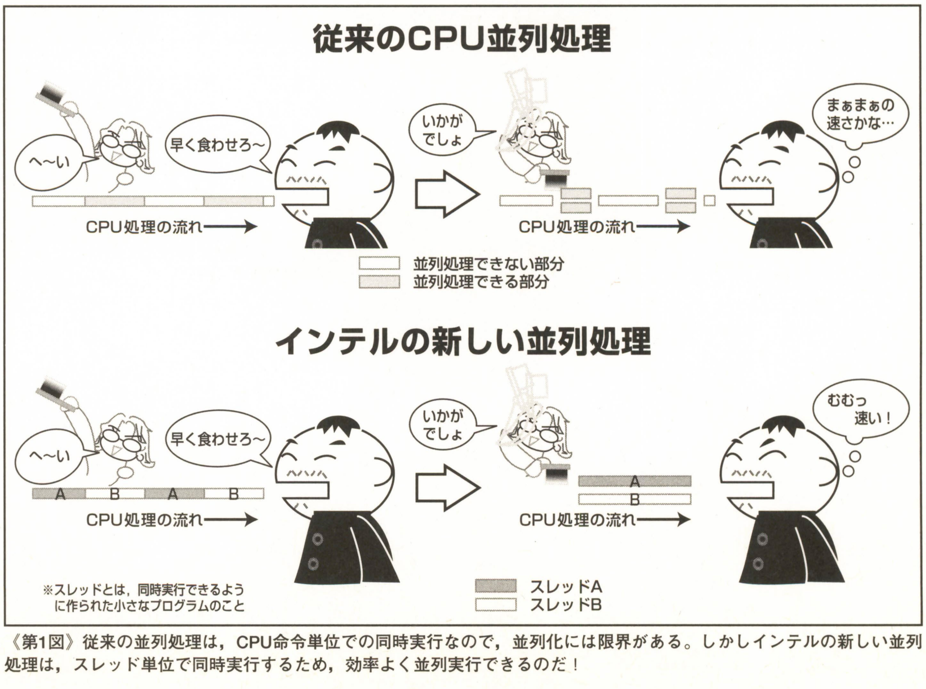
最近，高速でインターネットに接続するパソコンが増えてくることによって，ウイルスの感染速度が加速されているようです。

Windows NT／2000を利用しているかたは，Windows NT／2000サーバ上のIIS（Webサーバ）から感染する『コードレッド』というウイルスに，十分注意してください。

また，電子メールを使って感染するウイルスも非常に多く，ハードディスク内にある文書ファイルをウイルス化して送信する『SirCam（サーカム）』というウイルスの被害も多くなっています。

添付ファイルの送受信をするときには，おたがいにあらかじめ連絡を取り合って，まちがってもウイルスの成長の手助けをしないよう気をつけましょう。

●予想：秋葉原に少し行列ができる



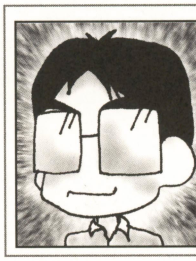
《第1図》従来の並列処理は，CPU命令単位での同時実行なので，並列化には限界がある。しかしインテルの新しい並列処理は，スレッド単位で同時実行するため，効率よく並列実行できるのだ！

プログラミング PROGRAMMING

プログラムの疑問・質問にお答えします

GET&RUN(ゲット ラン)

TEXT&PROGRAMMING●山崎 寛



『M5』や『MSX』でプログラミングを覚えた，BASIC歴15年の男。先日，普段使っているPDAがフル・リセット状態になってしまった（バックアップ電池の消耗が原因と思われる）。本体内にデータはほとんど入れていないため，たいした被害はなかったものの，それでも半月ぶんのちょっとしたデータが失われて少しショック。これに懲りて，これからはまめにバックアップを取ろうと思うが…いつまで続くやら。

アルゴリズム全般

VB

Q 僕の開発環境は，『Microsoft(マイクロソフト) Visual(ビジュアル) Basic(ベーシック)（以下VBと略）6.0』+『Direct(ダイレクト) Draw(ドロー)』です。これを使って，昔のPC-9800シリーズのパソコンで動作した，つぎのようなゲームを作りたいと思っています。

● **ブロックくずしのブロックの配置を，誰でも簡単にエディタ（メモ帳など）で作れるようなゲーム**

しかし，どうしてもテキスト・データを1行ずつ読み込むことができません。どうすれば実現できますか？

また，僕の作るブロックくずしは，かなりボールが大き目です。どうしても，ブロックとボールの当たり判定がおかしくなってしまいます。どうすれば，円と四角形の当たり判定を高速に行えますか？

【神奈川県・**古城公佑14歳**】

A 質問がいくつかわかれているので，一つずつ答えていきましょう。

VBでテキスト・ファイルからデータを読む

古城さんがどんな方法をためしたのかわかりませんが，VBでテキスト・ファイルを読み書きする方法の一つに，「OPENでファイルを開き，INPUTやPRINTで読み書きする」という方法があります。この方法を使ってデータを読み込むプログラムを紹介することにしましょう。

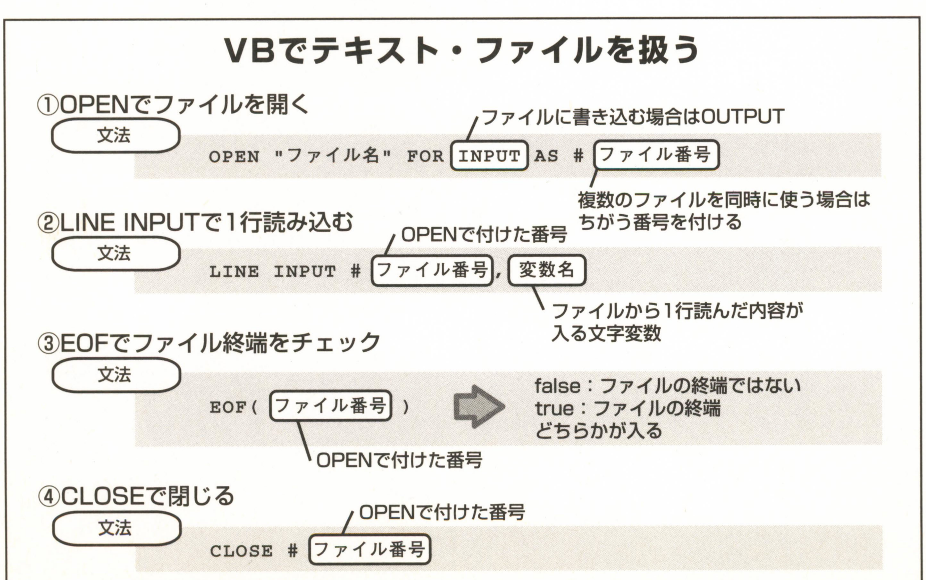
テキスト・ファイルを読み込むには，**第1図**のような手順になります。

まず，OPENでファイル名とファイル番号を指定します。このファイル番号を使いわけることで，複数のファイルを同時に読み書きできます。

ファイルを開いたら，LINE INPUTでファイルの内容を変数に代入します。LINE INPUTならば，テキストを1行ずつ読み込んでくれるので，古城さんの用途にはぴったりでしょう。また，ファイルの大きさが未定な場合などは，必要に応じてEOFでファイルの終わりをチェックします。

最後に，処理が終わったらCLOSEでファイルを閉じます。これで，ファイル読み込み処理は終了しました。

しかし，読み込んだデータはそのままでは単純な1行の文字列なので，これを



《第1図》VBでは，テキスト以外でもファイルを扱うのは簡単なのだ

《リスト1》テキストファイルからデータを読み込むサンプル（VB用）

```
Const bx = 19, by = 9'ブロックの横縦の数

Dim block(bx, by) As Integer

Private Sub Form_Load()

  Open "data.txt" For Input As #1
  For j = 0 To by
    Line Input #1, a$
    For i = 0 To bx
      block(i, j) = Val(Mid$(a$, i + 1, 1))
    Next i
  Next j
  Close #1


End Sub
```

《リスト2》リスト1で読み込む“data.txt”の内容（テキスト・ファイル）

```
11111111111111111111
11111111111111111111
11111111111111111111
11111111111111111111
11111111111111111111
11111111111111111111
11111111111111111111
11111111111111111111
11111111111111111111
11111111111111111111
```

処理する必要があります。おそらく，ブロックくずしのブロックのあり／なしや，種類などを配列変数に入れて管理しているでしょうから，MID＄で文字列を1文字（1文字＝1ブロックの場合）ずつ区切って配列へと入れていけばいいでしょう（**第2図**）。

テキストや配列内で，実際にどのようにデータを持つかについては，人それぞれ作りかたによって変わってくると思います。基本的には，このような形になるでしょう。

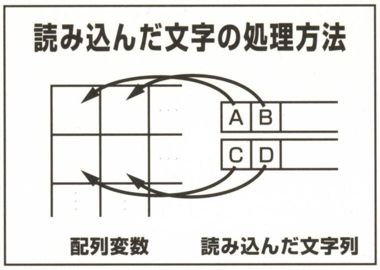
円と四角の衝突判定はどうする？

つぎに，円と四角形の衝突判定について紹介しましょう。

ゲームなどの速度が求められるプログラミングでは，多少見た目がおかしいのは無視して，円を四角形として扱ってしまう場合が多いと思います。

それでも判定が気になる場合，**第3図**のように円をいくつかの四角形の集合として，何回かにわけて四角形同士の衝突判定をするという方法があります。

高速な判定としてはこれでいいと思い

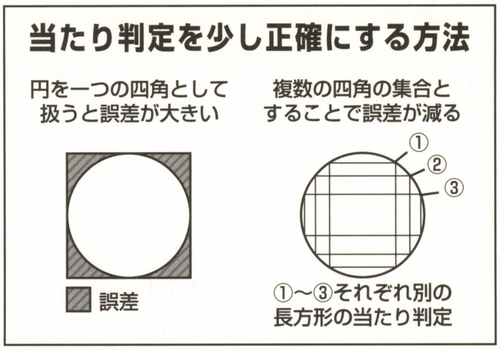


《第2図》このように1文字ずつ区切って，配列に入れていくのだ

ますが，さらに厳密な衝突判定を行いたい場合，つぎのような方法もあります（厳密な判定といっても，真円と長方形の場合それほどむずしい話ではない）。

第4図を見てください。これは，円の中心のX座標が四角の横幅に収まる場合です。この場合，単純に円の中心Y座標と四角の表面との差をとって，これが円の半径以下なら衝突します。これは，通常の長方形同士の衝突判定とほとんど変わらないので，わかりやすいでしょう。また，縦と横（XとY）を入れ替えても同じなのは言うまでもありません。

つぎに，**第5図**のように中心のX座標が四角から外れている場合ですが，この場合はY座標と表面の差だけではわかり



《第3図》円を，いくつかの長方形の集合として判定する。長方形が多いほど正確になるわけだ

ませんね。そこで，円に一番近い四角の頂点から円の中心までの距離を計ってやれば，この距離が円の半径以内なら衝突とわかるようになります。

（X1，Y1）と（X2，Y2）の2点間の距離を求めるには，

$\sqrt{(X1-X2)^2+(Y1-Y2)^2}$

とすればいいので，これで四角と円の座標から求めることができます。

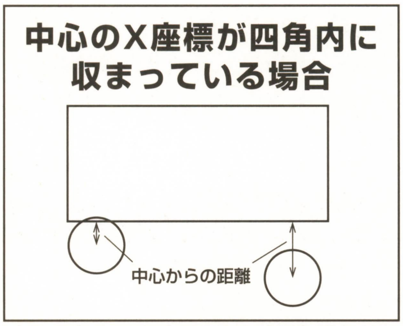
実際にVBのプログラムにしたものが，**リスト3**です。マウスで円を動かし，画面上の四角と衝突した場合にダイアログを表示します（**写真1**）。

グラフィックス作成

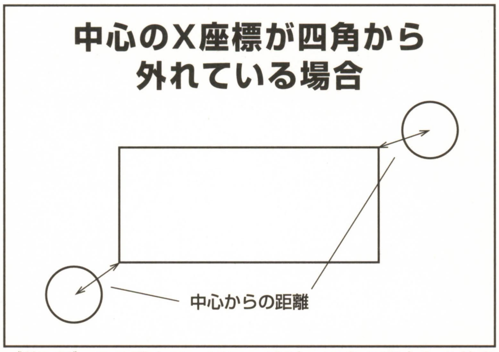
Q はじめての質問です（今年で購読4年目ですがまだまだわからないことといっぱいです）。

はじめての質問なのに，直接プログラミングのことではないことをおわびします。もしかすると，山崎先生の範囲外かもしれないですが，もしわからなければ他の先生にお願いしてください。どうしても知りたいことなのです。

Windows（Win32）用のゲームソフト制作を1人でやっていきたいのですが，ゲームの素材となるグラフィックス（今回は2D）の効率的な，または格好いい作成方法などというものを教えていただければうれしく思います。



《第4図》円の中心と四角の表面の差を計算し，円の半径以下なら衝突と判定する。単純に座標の差を計算しているわけだ



《第5図》この場合は，円に一番近い四角の頂点との差を計算する。円の半径以下なら衝突だ

《リスト3》円と四角を厳密に衝突判定するサンプル（VB用）

```
Const r = 100 '円の半径
Const bx1 = 240, by1 = 100, bx2 = 380, by2 = 200 '四角の座標
Dim rx As Integer, ry As Integer '円の座標

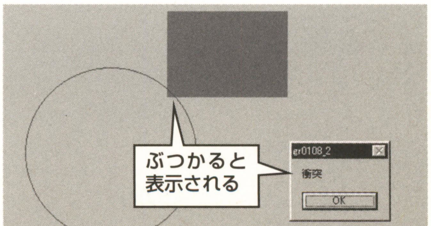
Private Sub Form_Load()
  Me.ScaleMode = 3
  Me.Width = 640 * Screen.TwipsPerPixelX
  Me.Height = 400 * Screen.TwipsPerPixelY
End Sub

Private Sub Form_MouseMove(Button As Integer, Shift As
Integer, X As Single, Y As Single)
  rx = X: ry = Y '円の中心をマウスカーソルの場所に合わせる
  flag = False '衝突判別用フラグ

  '衝突しているか厳密に調べる
  If rx > bx1 And rx < bx2 Then
     If Abs(ry - by1) < r Or Abs(ry - by2) < r Then flag =
True
  End If
  If ry > by1 And ry < by2 Then
     If Abs(rx - bx1) < r Or Abs(rx - bx2) < r Then flag =
True
  End If
   If Sqr((rx - bx1) ^ 2 + (ry - by1) ^ 2) < r Then flag =
True
   If Sqr((rx - bx1) ^ 2 + (ry - by2) ^ 2) < r Then flag =
True
   If Sqr((rx - bx2) ^ 2 + (ry - by1) ^ 2) < r Then flag =
True
   If Sqr((rx - bx2) ^ 2 + (ry - by2) ^ 2) < r Then flag =
True

  Me.Refresh
  If flag = True Then MsgBox "衝突"
End Sub

Private Sub Form_Paint()
  Line (bx1, by1)-(bx2, by2), QBColor(2), BF
  Circle (rx, ry), r, QBColor(4)
End Sub
```



《写真1》リスト1の実行結果。このサンプルでは，衝突を厳密に調べているぞ

使用するソフトは，『Adobe Photoshop（以下Photoshopと略）5.0J』です。ちなみにゲーム制作では，『Microsoft Visual C++6.0』と『DirectX 8.0a SDK』を使っていくつもりです。

もう少しくわしく説明します。インターネットで入手できる，フリーソフトのゲームがありますよね。実名をあげるとたとえば『BM98』とか，とても一般の人が作ったとは思えないようなグラフィックスを用いているゲームがあります。そのフォルダの中身をのぞいてみると，何やら“～.bmp”というファイルがあり，それを開くとゲーム中で見たグラフィックスが入っているではないですか。

たとえば「タイトル・ロゴ」「Game Start」などの文字について，その項目を選択したら大きくなって明るい表示，別な部分を選択したら小さくなって暗くなる，などはどう処理しているのでしょうか。

また，一番知りたいのは，RPGの「主人公や町の人」の移動中グラフィックス，フィールドや街，城，木などのマップ用グラフィックス，敵キャラクターのグラフィックスなどの管理です。これは，どうやっているんでしょう？。

ついでといっては失礼ですが，一般に販売されているゲームでは，こういったグラフィックスは「見えない」ようになっていますよね。圧縮しているのであれば，参考までにそれらによく使われている圧縮アルゴリズムも教えていただけないでしょうか。簡単なハフマン圧縮やランレングスのアルゴリズムはわかります。

それらの管理方法，実際の利用，などなどを教えてください。

また，ゲーム作成にうってつけな本があれば，紹介していただけないでしょうか。

【東京都・Infinite Creator18歳】

A はじめての質問ということですが，もう少し質問内容を整理して，知りたい点を明確にしてもらえると助かります。

まあそれはいいとして，グラフィックスの作成方法については，おっしゃるとおり私は範囲外です。手近にいてくわしそうな，かいぬまたにし先生に聞いてみました。

結論からいうと，効率的に絵を描いたり，誰でもすぐに格好よいグラフィックスを作成できるような方法はなく，地道にたくさん絵を描いて練習するのが早道なのではないか，ということでした。

Infinite Creatorさんが，もし人物や建物など，特定のものの描きかたを習得したいのであれば，それなりの上達法もあるでしょう。しかし，ゲームの世界すべてのグラフィックスを描きたいのであれば，あらゆるものの観察とスケッチからはじめる他はないでしょう。

また，Photoshopのテクニックや使いかたについては，製品付属のマニュアルをしっかり読み，チュートリアルをひととおりためしてみるだけでもかなり身につきます。市販の書籍を参考にするのもよいのですが，まずはここからはじめてみるのもいいですよ。

市販や個人作成のゲーム，それからインターネットで公開されている素材画像などをよく観察することで，自分の描いたものと比較して「どうして格好よく見えるのか」などを考えてみるのも一つの手でしょう。

タイトルロゴなどの文字については，ゲームに限らず，最近ではホームページでもそうしたロゴや画像処理した文字などを使うことが多いでしょう。ホームページ作成のソフトや解説なども参考になると思います。

ロゴなどについて，より本格的に取り組みたい場合には，『Adobe Illustrator』などのベクトル・データを扱うソフトを用いることになります。

2D画像を描く場合，光と影を表現することにより立体感が生まれ，リアリティが増します。光源の方向などを考えながら，陰影を付けてみるといいでしょう。

メニューの文字などを選択したら，文字が明るくなったり暗くなったりする処理は，あらかじめそれぞれの画像として用意してもかまいません。しかし，最近ではパソコンの能力向上もあり，プログラムによって実行中に計算で描く場合が多いようです。表示する1ドットごとについて，輝度や位置に定数を乗算や除算することで，明るさや大きさを変化させることができます。

RPGなどでたくさんのキャラクター画像を管理する方法についてですが，データを作成する際になるべく規則性をつけることで管理しやすくなります。たとえば，人物だったら方向や動作によっていくつかの画像を用意しますが，これらの順番などを共通にしておくことで，画像を取りだすプログラムを共通化できます。また，画像の種類によって複数のファイルに入れる場合などは，ファイル名のつけかたに規則があれば扱いやすくなるでしょう。

市販ソフトの画像格納方式ですが，市販ソフトといってもさまざまなものがあり，一概に「この方式がいい」とはいえません。実際のところ，BMPやPNG，JPEGといった標準的なファイル形式や，無圧縮のベタなデータのまま収録しているソフトも多くあります。

圧縮を独自に行う場合，可逆圧縮の場合は，“LZ77法”やこれを2次元に当てはめた方式があります。また，非可逆圧縮の場合は，輝度と色に分割して離散コサイン変換といったJPEGと同様な方式を使うのが標準的です。画像圧縮の詳細については，画像ファイルのフォーマットに関する仕様書や画像処理に関する文献などを参考にするといいでしょう。

駆け足で回答しましたが，また疑問点があったら質問してください。それでは，また来月。

疑問・質問ないですか？

このコーナーでは，プログラムに関するQ&Aを掲載しています。

プログラムに関する疑問（「こんなゲームを作りたいんだけど，ここがうまくいかない！」とか「このプログラムの入力方法がわからない～」）などがありましたら，なるべく具体的に，使用機種や言語を明記して，下記住所までハガキや封書もしくは電子メールでお送りください。住所，氏名（ペンネーム），年齢，職業，電話番号も忘れずに。

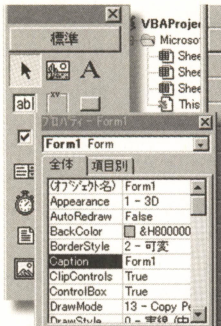
■郵便あて先

〒141-8715 東京都品川区東五反田1-11-15 電波新聞社マイコンBASICマガジン編集部 GET&RUN係

■電子メールあて先

bm@jolls.co.jp

※メールのタイトル／Subjectに「GET&RUN係」と明記してお送りください。



はじめよう Microsoft ExcelでのBASICプログラミング

協力 富士通南九州システムエンジニアリング

第7回 パズルゲームをつくってみよう

この連載では，Microsoft Excelを使って，BASICのプログラミングを行います。今月からゲーム・プログラミングをテーマに，簡単なゲームを作成しながらプログラミングの説明をおこないます。

TEXT & PROGRAMING 椛島たけし



Dr.Dの得意技、指コンは今でもつかう場合がある。そんなとき，昔必死で練習したグワシ（後期型）が役に立つ。グワシとは中指と小指だけを曲げる技なのだ。

Excel VBAでのゲームとは?

プログラミングを始めるきっかけは「ゲームを作りたい」というケースがほとんどなのではないでしょうか？Excel VBAは，わざわざプログラミング用のソフトを購入しなくても，ほとんどのWindowsパソコンにバンドルされているExcelで，プログラミングできるのが魅力です。

しかし，純粋なプログラミング・ツールではないので，目的とするゲーム・プログラムが100%作成できるわけではありません。ただ，ゲームの種類によっては，かなり魅力的なゲーム・プログラムを作成できます。まずは，Excel VBAで作成できるゲーム・プログラムを探ってみましょう。

Excel VBAでのゲーム・プログラムとは?

Excel VBAでゲーム・プログラムを作成する場合のプラス面とマイナス面を考えてみましょう（第1図）。

Visual BasicやVisual C++などに比べ，Excel VBAの機能は限定されています。しかし工夫しだいでは魅力的なゲーム・プログラムを作成することが可能です（ベーマガに投稿されているプログラムが物語っていますね）。

まずは，Excel VBAで作れそうなゲーム・プログラムを考えてみましょう（第2図）。

●シューティング・ゲーム

弾丸でエイリアンを撃ち落とすようないわゆるシューティング・ゲームは，高速な描画（画面表示）が必要です。したがって，Excel VBAは苦手です。同じ理由で，格闘系のアクション・ゲームも苦手です。テトリスのような落ちモノもほぼ同様です。

●RPG（ロールプレイング・ゲーム）

Excelのセルなどを上手く利用することで，単純なRPGは作成可能です。しかし，凝ったキャラクターの動きは苦手です。

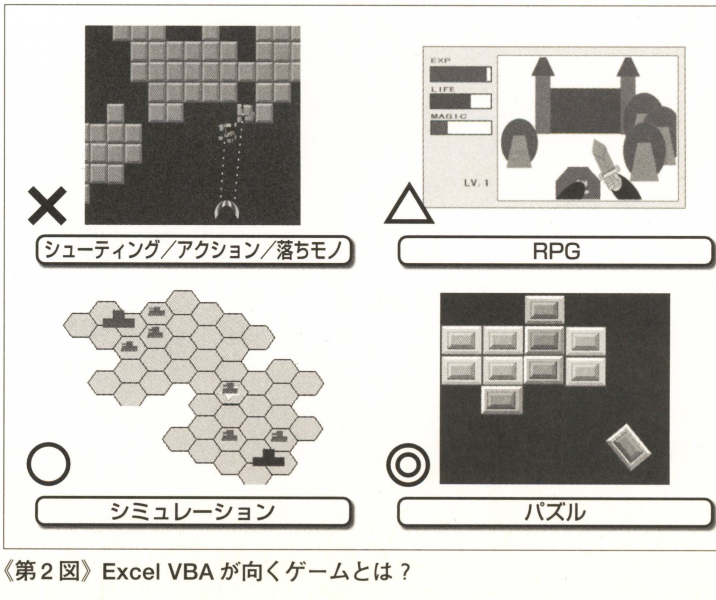
プラス面

- Basicで比較的やさしくプログラミングできる
- Excelのワークーシートを利用できる
- 図形（Shape）などの部品を利用できる
- グラフで結果表示などが簡単にできる

マイナス面

- タイマーコントロールがないため時間系列を伴う処理が苦手
- 動作速度が速くない
- 実行形式のプログラム（EXE）を作成できない

《第1図》Excel VBAの特徴



《第2図》Excel VBAが向くゲームとは？

6	1	8
7	5	3
2	9	4

縦，横，斜めの合計がすべて15

《第3図》魔方陣は縦，横，斜めの合計がすべて同じになる

●シミュレーション・ゲーム

Excelはもともと計算やデータ処理用のソフトウエアですので，予測計算が必要なシミュレーションは向いています。シミュレーション・ゲームの場合は高速な画面表示などはありませんので，向いているジャンルと言えるでしょう。

●パズルゲーム

アニメーション表示や高速な画面表示の必要がないパズルゲームは，Excel VBAの得意種目です。プログラム·リストも意外に単純化できます。実際，ベーマガに投稿されるExcel VBAプログラムの多くがこのジャンルですね。

パズルゲームを検討すると

今回はパズルゲームに絞ってゲームの作成を考えてみましょう。まず，パズルゲームには大きくわけて2種類あります(と筆者は思っています)。

●正解を求めるゲーム

いわゆる一種のクイズです。したがって当然正解が必要です。ジグソーパズル，シルエットクイズなど一般的なゲームをテーマにしたり，自分なりのアイデアしだいで，いろいろなゲームが作成できそうです。

●思考を行うゲーム

対戦型でターン（攻守の切り換え）があるタイプのゲームです。コンピュータ対戦型とすると必然的に思考ルーチンが必要となります。チェス，オセロ，将棋，囲碁などコンピュータの「強さ」をうたい文句に商品化されているものも多いですね。人間とチェスが対戦し勝ち負けを競うなど，「思考」は奥が深い分野です。

コンピュータ対戦型のゲームはなかなか魅力的です。思考ルーチンを作成するには，作成するゲームについて，定石と呼ばれる一般的な「手」を知ることが必要となります。この定石を考えてプログラミングするのはちょっとホネですので，はじめは，「正解を求めるゲーム」のほうが作成しやすいと思います。

今回は「魔方陣」と呼ばれる比較的おなじみのパズルをゲームにしてみます。

魔方陣とは？

魔方陣とは，縦横同数の方眼があり，縦横の合計（場合によっては縦横斜めの合計）がすべて同じ数となるものです（第3図）。数学の授業などで見たことがあるかもしれませんね。

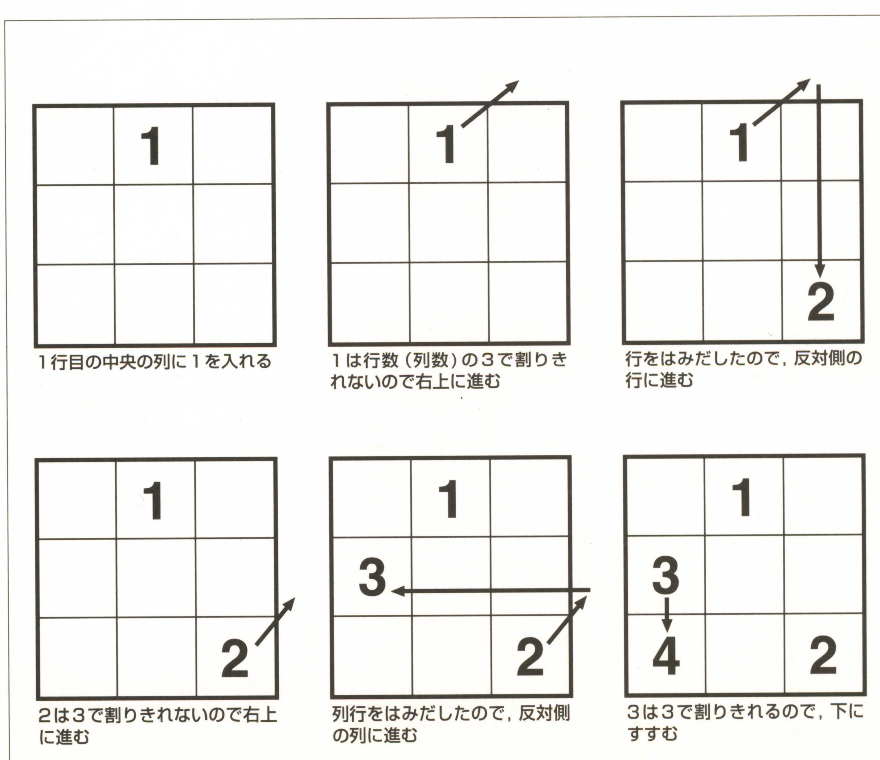
魔方陣のなかでも，奇数魔方陣（方陣数が奇数のもの）は，解を求める比較的簡単な方法があります。そこで，この方法を利用したゲーム・プログラムを作成してみることにしましょう。

奇数魔方陣を解くアルゴリズム

まずは奇数魔方陣を解く方法，つまり正解が必要です。奇数魔方陣は，つぎの方法で解くことができます。第4図といっしょに見てください。

① N個×N個の空白の方陣を作成する。
② 右上に進むか左上に進むか決める(以下，右上として説明)
③ 1行目の中央の列に1を入れる(セルに入れた数字をXとする)
④ XがNで割り切れる場合，下に進む。割り切れない場合，右上に進む
⑤ 方眼がない場合は反対側の列(行)に移動する
⑥ セルにXに1を足した数字を入れる。(セルに入れた数字をXとする)
⑦ XがN×Nより少ない場合，④に戻る。等しい場合終了する。

それでは，方陣にExcelのセルを利用することを想定して，流れ図を作成してみます(第5図)。流れ図にすると，アルゴリズムが明確になり，プログラムが作成しやすくなります。

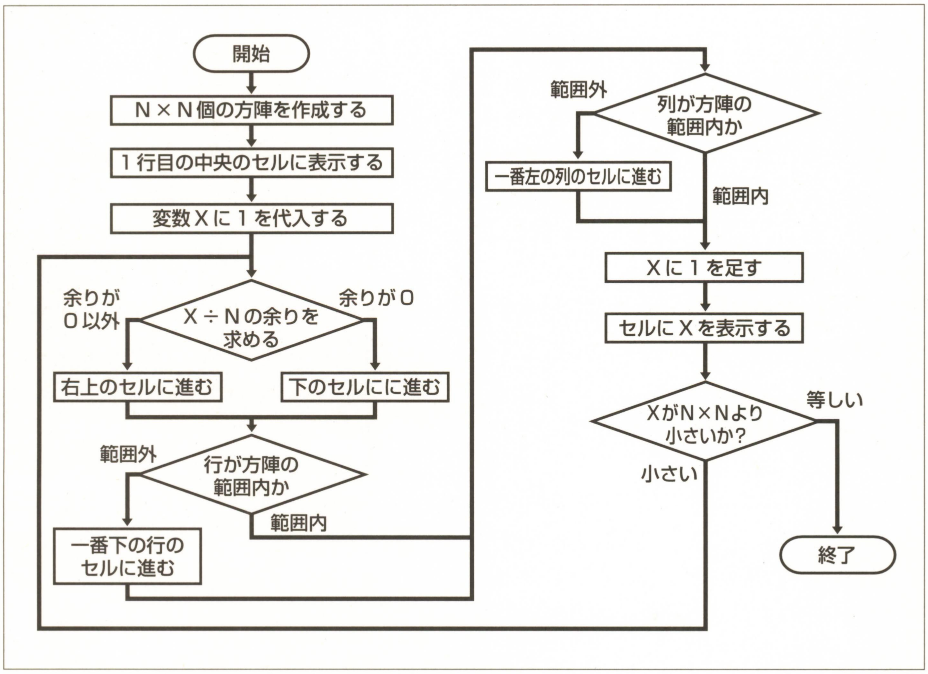


《第4図》奇数魔方陣を解くアルゴリズム

Excel VBAで奇数魔方陣を解く

Excelのセルを利用して奇数魔方陣の正解を求めるプログラミングをはじめましょう。商と剰余を求めるため見慣れない関数がありますが，流れ図のとおりにプログラミングしたのがリスト1です(実行結果は第6図)。

意外と短いプログラムですが奇数であれば何桁でも正解を表示することができます。

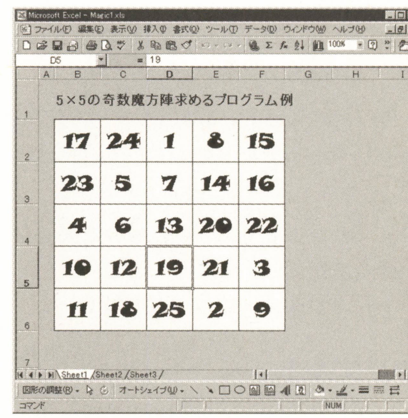


《第5図》奇数魔方陣を解くアルゴリズムを流れ図にする

ゲーム化を考えよう

さてつぎはゲーム化することにします。出題形式となりますので，確実な正解が必要ですが，じつは奇数魔方陣の答えは8とおりあります（第7図）。二つ以上ヒント（答え）を表示すると必ず上下左右が確定し，正解は一つとなります。

ほかにもゲーム化するには，いろいろな機能を組み込む必要があります。



《第6図》プログラムで奇数魔方陣を解いた実行結果

8	1	6
3	5	7
4	9	2

4	9	2
3	5	7
8	1	6

8	3	4
1	5	9
6	7	2

6	7	2
1	5	9
8	3	4

6	1	8
7	5	3
2	9	4

2	9	4
7	5	3
6	1	8

4	3	8
9	5	1
2	7	6

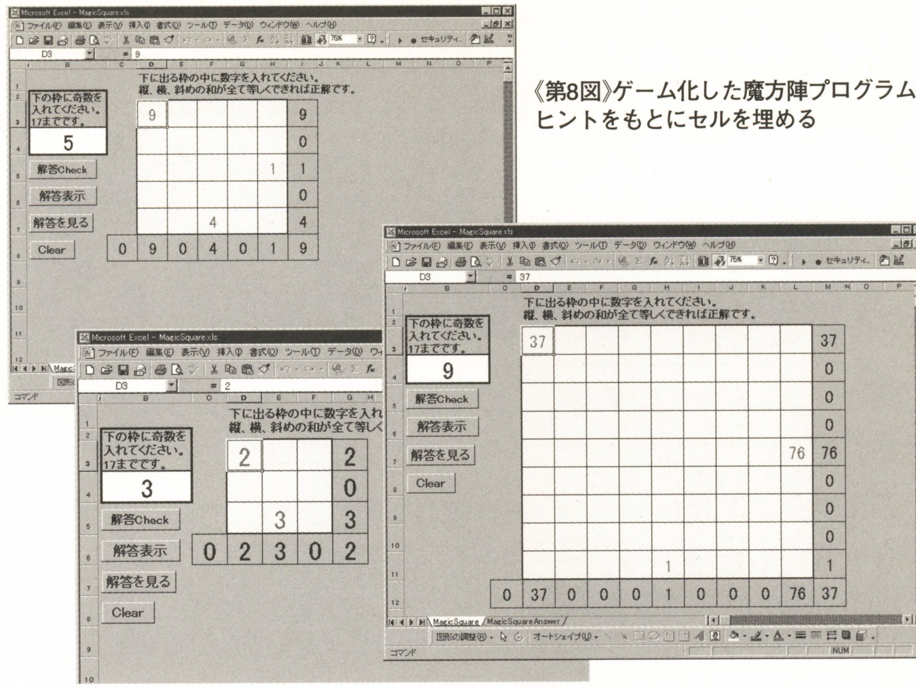
2	7	6
9	5	1
4	3	8

任意の二つの数が特定できればどのパターンか判別可能

《第7図》奇数魔方陣には8とおりの正解がある

①方眼の数の入力
②ヒントの表示
③正解の表示
④答え合わせの機能
⑤入力された数字のチェック機能

このような機能を組み込んだプログラムを作成してみました(第8図)。リストは長いので，インターネットでダウンロードしてください。



《第8図》ゲーム化した魔方陣プログラム。ヒントをもとにセルを埋める

いろいろなパズルゲームを作成しよう

今回からしばらくパズルゲームに的を絞り，ゲーム・プログラムのアルゴリズムや思考ルーチンを説明，紹介していきたいと思います。

それではお楽しみに。

《リスト1》奇数魔方陣を解くプログラム・リスト

```
Dim lngN As Long                '行・列の数
Const lngRowStart As Long = 2   '方陣の開始行
Const lngColStart As Long = 2   '方陣の開始列

Public Sub StartGame()
    Dim lngRowNow As Long    '表示するセルの行
    Dim lngColNow As Long    '表示するセルの列
    Dim lngCellData As Long  '表示するデータ

    lngN = 5                  '行データ・列データを5とする
    lngCellData = 1            '始めに表示するデータ
    lngRowNow = lngRowStart '始めに表示する行(先頭行)

    '始めに表示する列(中央を求める)
    lngColNow = lngColStart + (lngN ¥ 2 + 1) - 1

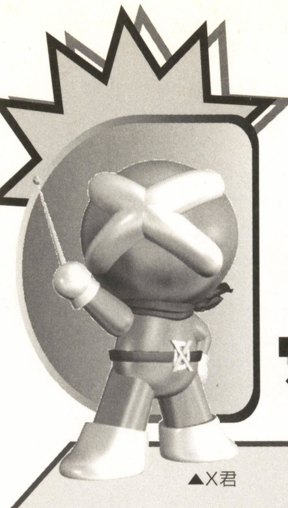
    'セルにデータを表示
    Sheet1.Cells(lngRowNow, lngColNow) = lngCellData

    'セルの数だけ繰り返す
    Do Until lngCellData = lngN ^ 2
        If lngCellData Mod lngN = 0 Then '行データで割り切れる場合
            lngRowNow = lngRowNow + 1   '下のセルを指定(行+1、列はそのまま)
            lngColNow = lngColNow
        Else                             '割り切れない場合
            lngRowNow = lngRowNow - 1   '右上セルを指定
            lngColNow = lngColNow + 1
        End If

        Select Case lngRowNow   '移動先の行を確認
        Case Is < lngRowStart   '行が範囲外か確認
            lngRowNow = lngRowStart + lngN - 1  '一番下のセル(行)にを指定
        End Select

        Select Case lngColNow
        Case Is >= lngColStart + lngN '列が範囲外か確認
            lngColNow = lngColStart '一番左のセル(行)にを指定
        End Select

        lngCellData = lngCellData + 1 'データに1を足す
        Sheet1.Cells(lngRowNow, lngColNow) = lngCellData 'セルにデータを表示
    Loop
End Sub
```



▲X君

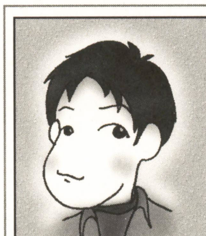
マイクロソフトの技術者がDirectXテクノロジーを伝授する！

IE+DirectAnimation マルチメディア・プログラミング

第8回　ピッカブル・イベントを使って“あみだくじゲーム”を作ろう！

『Microsoft Internet Explorer（以下IEと略）』と『DirectAnimation*1』を使って，誰でも簡単マルチメディア・プログラムを作れるテクニックを紹介しています。今回は，前回に引き続き“イベント*2”に関するテクニックを取り上げ，さらに軌跡に沿ったイメージの移動という新しい変換についても紹介します。

TEXT&PROGRAMING●平井昌人



横浜市生まれの＆H25歳男性。1993年4月に，開発言語製品のサポート・エンジニアとしてマイクロソフトに入社。現在はシステム・エンジニア（SE）としてVisual Studioを中心とした開発系の啓蒙・支援活動を行っている。アメリカの同時多発テロのおかげで，ロサンジェルス出張が取り消された…。

＊1　DirectAnimation
2Dや3Dグラフィックス，サウンド，映像などを総合的にサポートする高レベルなAPIを提供している。IEに統合されており，インターネットでの利用を強く意識した設計となっている。

＊2　イベント
Windows上で起こるあらゆる“出来事”を総称していう。マウスが動いた／マウスがクリックされた，キーが押された／離された，指定した時間が経過したなどが代表的なイベント。他にも，ウィンドウが閉じたとか，メールが着信したなどの事象もイベントといえる。サンプル・コードにある「Sub Window_OnLoad()」や「Sub Window_OnResize()」などをイベント・プロシージャといい，特定のイベント（ロードされた／ウィンドウのサイズが変わった）が発生すると，そのプロシージャが実行される。

今回のテーマは二つある！

先月号では，マウスのイベントに反応させるテクニックを取り上げ，インタラクティブな（相互に作用する）アニメーションを紹介しました。イベントを使うことで，ユーザーの操作に反応する複雑で多彩な動きや変化を組み込むことができます。

今月も，引き続きイベントに関するテクニックを取り上げます。また，軌跡に沿ったイメージの移動という新しい変換についても紹介します。今回のテーマは，つぎの二つです。

●軌跡に沿って図形や画像を移動させる

●図形や画像の部分だけに反応するイベント（ピッカブル・イベント）

そして，この二つのテクニックを応用して，**写真1**のようなゲームを作ってみましょう。

軌跡に沿って図形や画像を移動させる方法

はじめに，軌跡（DAPath2オブジェクト）についての復習と，新しく登場する「軌跡に沿って移動させる」というテクニックを解説しましょう。

サンプル・プログラム『くるくる天使』と『てんとう虫』の2本を用意しました。本文と合わせて動作を確認してください。ソースコードは，**リスト1**と**リスト2**です。サンプルを実行すると，それぞれ**写真2**と**写真3**のように，キャラクターが軌跡にそって移動します。

軌跡とは「図形のもとになる型紙」

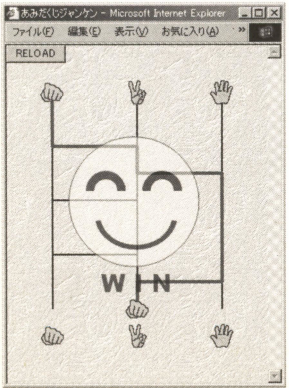
軌跡を表すDAPath2オブジェクトは，非常に利用頻度の高いオブジェクトの一つです。軌跡とは，ズバリ「図形のもとになる型紙」です。

2次元座標上には，線，三角形，四角形，円，楕円，扇形，星型などいろいろな図形があります。これらを描くために，型紙があると便利ですよね。まさしくこの型紙が，軌跡（DAPath2オブジェクト）なのです。

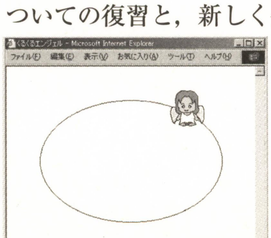
軌跡の定義を知っておこう！

軌跡は，Ovalメソッド（円），Rectメソッド（長方形），StringPathメソッド（文字列）などを使って定義します。また，いくつかのDAPath2オブジェクトを組み合せることで，より複雑な図形や非幾何学的な形まで定義することもできます。

つぎに，軌跡を定義する例をいくつか示して



《写真1》今回作るサンプル『あみだくじジャンケン』



《写真2》これはFollowPathを使ったサンプル『くるくる天使』の実行画面



《写真3》FollowPathAngleを使ったサンプル『てんとう虫』の実行画面がこれ

おきます。

●Ovalメソッド

幅300，高さ200の楕円の軌跡を定義します。**写真2**のサンプルで利用しています。

```
Set pth = m.Oval(300, 200)
```

●StringPathメソッド

フォント“Arial”，12ポイントの書式で“ABC”という文字列の軌跡を定義します。

```
Set fnt = m.Font("Arial", 12, m.Red)
Set pth = m.StringPath("ABC", fnt)
```

●Polylineメソッド

点（DAPoint2オブジェクト）の配列にしたがって，星型の軌跡を定義します。開始点と終了点がちがうと図形として，閉じていない状態になります。このような場合は，Closeメソッドで閉鎖矩形にできます。ただし，線だけでよい場合，Closeメソッドは不要です。

Polylineメソッドは，**写真3**のサンプルと，後で紹介するサンプル・ゲームで利用しています。

```
Dim xy(9)
Set xy(0) = m.Point2(0, 140)
Set xy(1) = m.Point2(-36, 32)
Set xy(2) = m.Point2(-152, 32)
Set xy(3) = m.Point2(-56, -36)
Set xy(4) = m.Point2(-96, -144)
Set xy(5) = m.Point2(0, -76)
Set xy(6) = m.Point2(96, -144)
Set xy(7) = m.Point2(56, -36)
Set xy(8) = m.Point2(152, 32)
Set xy(9) = m.Point2(36, 32)
Set pth = m.Polyline(xy)
Set pth = pth.Close
```

軌跡からイメージへ！

DAPath2オブジェクトのDrawメソッドまたはFillメソッドで，その軌跡が表現する図形，つまりイメージ（DAImage）を作成します。このときに，枠線のスタイルを決定するDALineStyleオブジェクトや，塗りつぶしのイメージを決定するDAImageオブジェクトを事前に用意しておく必要があります。

●Drawメソッド

枠線（border）のみを描画します。

```
Set border = m.DefaultLineStyle
Set img = pth.Draw(border)
```

●Fillメソッド

枠線（border）と内部塗りぶし（panel）で描画します。

```
Set border = m.DefaultLineStyle
Set paint = m.SolidColorImage(m.Blue)
Set img = pth.Fill(border, paint)
```

軌跡に沿った移動～FollowPathメソッド／FollowPathAngleメソッド

ここが，今回のテーマの一つになっている部分です。

軌跡は，図形の描画のためだけでなく，移動の“道しるべ”として利用することができるのです。もちろん移動は，2次元座標での変換を表すDATransform2オブジェクトです。これまで紹介した平行移動，回転，伸縮などの変換に，「軌跡に沿って移動させる」という新しいレパートリーが加わります。

軌跡に沿って移動させる変換を定義するには，FollowPathメソッドまたはFollowPathAngleメソッドを使います。どちらも第1引数に定義済みのDAPath2オブジェクトを渡し，第2引数に移動にかかる秒数をあたえます。

つぎのコードは，天使のイメージ“CH210.GIF*3”を幅300，高さ200の楕円の軌跡に沿って，1周5秒で移動させます。

```
Set pth = m.Oval(300, 200)
Set tran = m.FollowPath(pth, 5)
Set img = m.ImportImage("CH210.GIF")
Set img = img.Transform(tran)
```

なお，このままだと1周して移動が完了してしまいます。移動を続けるのであれば，

```
Set tran = m.FollowPath(pth,
5).RepeatForever
```

というように，RepeatForeverメソッドを使います。

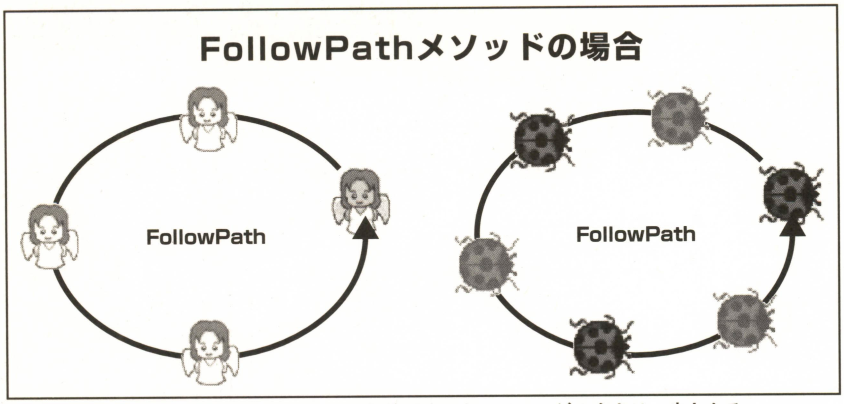
FollowPathメソッドとFollowPathAngleメソッドのちがいは，**第1図**と**第2図**に示してあります。右側は，画像ファイルをてんとう虫“TENTO.GIF”に変更したものです。

図形や画像の部分だけに反応するイベントを組み込む！

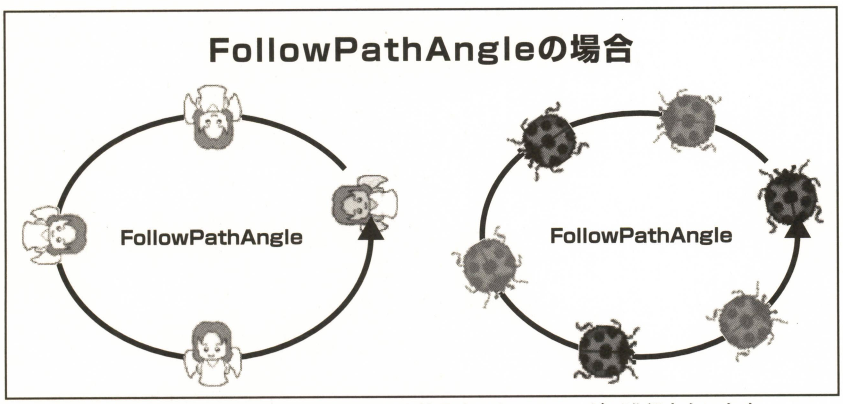
前回は，マウスのイベント*4（左ボタン・ク

*3　CH210.GIF
このアニメーションGIFファイル“CH210.GIF”は，ナツメ社『インターネットホームページ素材集Pack3000 GIFアニメーション＆サウンド編』から使用しています。

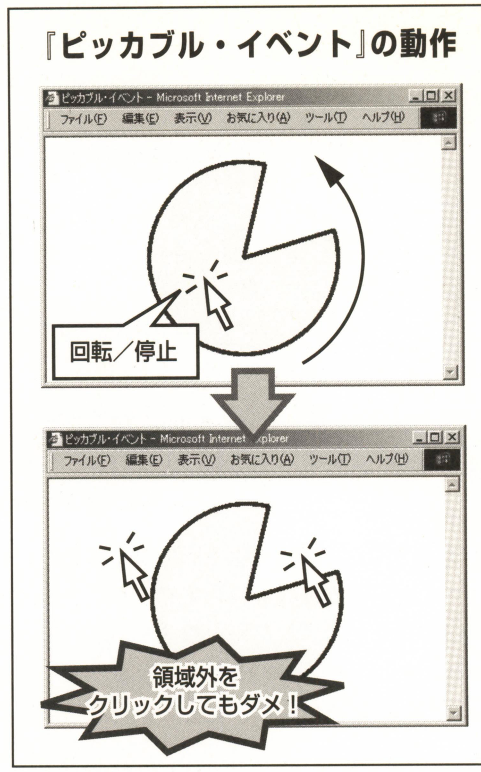
*4　マウスのイベント
Windowsでは，“クリック”とか“ダブル・クリック”とかいう操作が定義されているが，マウスという装置ではこの考えかたをしない。マウスのボタンが“押された”とか“離された”という，単純なイベントとしてしか認識しないのだ。つまり，クリックはマウスの左ボタンが押された（m.LeftButtonDown）というイベントに等しくなる。



《第1図》FollowPathメソッドを使用すると，イメージの向きは一定となる



《第2図》FollowPathAngleメソッドを使用すると，イメージが進行方向に向く



《第3図》図形内部をクリックすると反応するが，図形の部分以外では反応しないのだ

リック）を使ったインタラクティブなアニメーションを紹介しました。今回はさらに発展させて，イメージ（図形や画像）の部分だけに反応するイベントを組み込むテクニックを紹介します。

前回紹介したテクニックでは，画面上のどこをクリックしてもイベントが発生します。そこで，図形やキャラクターの部分をクリックしたときだけにイベントが発生するように改良してみましょう。

ここでは，簡単なサンプル・プログラム『ピッカブル・イベント』をもとに解説していきます。ソースコードは，**リスト3**です。このサンプルを実行すると，**第3図**のように中央で扇形がクルクルまわっています。扇形の部分をクリックする（マウスの左ボタンを押す）と，回転の開始／停止が制御できます。扇形の部分以外をクリックしても，反応しないところがミソです。

ピッカブルとは何か？

このように，図形や画像の部分だけに反応させるためには“ピッカブル*5”が必要です。ピッカブルは，イメージ（画像や図形）の有効な部分だけをピックアップしたものだと考えてください。そして，図形や画像の部分だけに反応するようなイベント（DAEventオブジェクト）を，“ピッカブル・イベント”と呼びます。

ピッカブルは，DAPickableResultオブジェクトで管理されます。DirectAnimationにおいてピッカブル・イベントは，非常に強力な武器となります。とくに，インタラクティブ性の高い複雑なアニメーションには，欠かせない機能といえるでしょう。

ピッカブルを利用すると，「あるキャラクターだけに反応するイベント」や「キャラクターごとに違ったイベント」を組み込むことができます。

ピッカブルの作成 ～Pickableメソッド

つぎのコードのように，イメージ（DAImageオブジェクト）のPickableメソッドでDAPickableResultオブジェクトを作成します。

Pickableメソッドは，戻り値としてDAPickableResultオブジェクトを返します。変数imgはDAImage型，picはDAPickableResult型の変数です。

```
Set pic = img.Pickable
```

ピッカブル・イベントの取得 ～PickEventプロパティ

ピッカブル上でのイベント（DAEventオブジェクト）をPickEventプロパティから取得します。

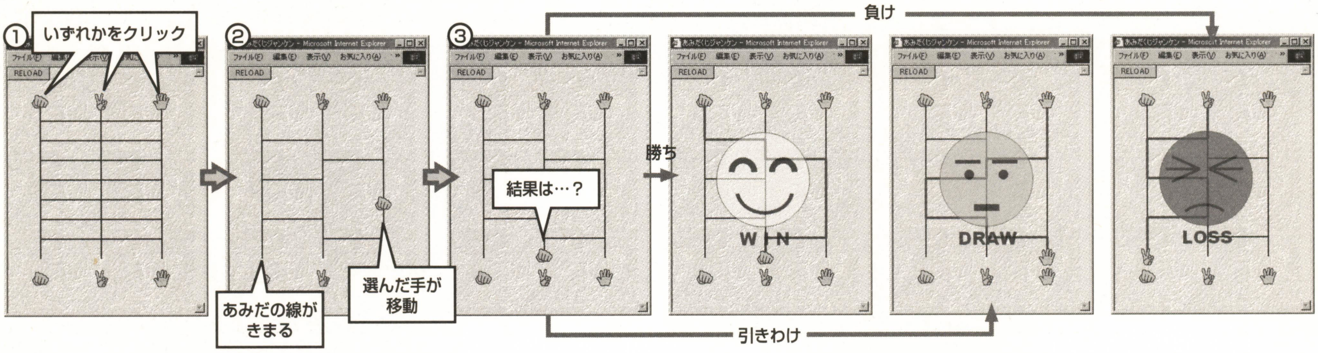
PickEventプロパティで取得したピッカブル・イベントは，「クリック」とか「キーが押さ

＊5　ピッカブル【Pickable】
うまくあてはまる日本語がないので，本稿ではそのままピッカブルと呼ぶことにする。ピッカブルは，Windowsでいうところの“リージョン【Region】”に似ている。ほとんどのウィンドウは四角形であるが，別に四角形である必要はない。丸いウィンドウでも，星型のウィンドウでもいいわけだ。このような形状を決定しているのがリージョンである。リージョンを自分で定義すれば，いろいろな形のウィンドウを表示することができる。例として，『Windows Media Player』や『Microsoft Office 2000』に付属するOfficeアシスタントがある。



▲Windows Media Playerは，ウィンドウ・デザインをいろいろ選択できるのだ

これが『あみだくじジャンケン』だ！



《第4図》今回は音楽や効果音を入れなかったが，腕に覚えのある人はつけてみよう。迫力がでるぞ！

れた」とかいう限定されたものでなく，その上で発生するすべての事象に対するものです。つまり，ピッカブルの上でのマウスの左クリック，右クリック，マウスの移動などをすべてイベントとして認識します。

```
Set ev = pic.PickEvent
```

イベントの組み合わせ ～AndEventメソッド／OrEventメソッド

PickEventプロパティで取得したピッカブル・イベントをそのまま利用することも可能ですが，そのピッカブル上のすべてのイベントに反応してしまうため，通常は特定のイベントに限定させなければなりません。

その方法は，AndEventメソッド*6やOrEventメソッド*7を用いて，ピッカブルのイベントと他のイベントをAndやOr条件で組み合わせるというものです。

つぎのコードは，ピッカブル・イベント（ev）とマウスの左クリック（m.LeftButtonDown）をAnd条件で組み合わせて，ピッカブル上でマウスの左クリックのみのイベントに限定しています。

```
Set ev = m.AndEvent(m.Left
Button Down, ev)
```

また，つぎのように定義すれば，ピッカブル上で「マウスの左右どちらか一方のボタンが押されたとき」発生するイベントとなります。

```
Set ev = m.AndEvent(m.OrEvent
(m.LeftButtonDown, m.RightButton
Down), ev)
```

ちなみに，つぎのように記述すると，どのようなイベントになるでしょうか？

```
Set ev = m.AndEvent(m.AndEvent
(m.LeftButtonDown, m.RightButton
Down), ev)
```

これは，ピッカブル上で「マウスの右左のボタンを両方同時に押したとき」に発生するイベントです。もちろん，AndEventメソッド，Or Eventメソッドは，ピッカブルでなくても利用できます。

ピッカブルからイメージを取得 ～Imageプロパティ

ピッカブルは，実際のイメージでありません。ピッカブルを目に見えるようにするためには，DAPickableResultオブジェクトのImageプロパティから再度，イメージ（DAImageオブジェクト）を取得しなければなりません。

ピッカブルのもとになったイメージ（DAImageオブジェクト）を表示しても，イベントが認識されないので注意しましょう。

```
Set img = pic.Image
```

サンプル・プログラム 『あみだくじジャンケン』で遊ぼう！

今回紹介した新しい二つのテクニック，「軌跡に沿った移動」と「ピッカブル・イベント」を応用して，サンプル・ゲーム『あみだくじジャンケン』なるものを作ってみました。

ソースコードは，**リスト4**です。なお，サンプルのソースコードや画像ファイルなどは，ベーマガ・ホームページにありますのでダウンロードしてください。

●http://www.basicmagazine.net/

それでは，遊びかたを紹介しましょう。

このサンプルは，名前のとおり，“あみだくじ”

＊6　AndEventメソッド
AndEventメソッドは，「AかつB」でイベント（DAEventオブジェクト）どうしの論理積を求める。「Set newEv = m.AndEvent(eventA, eventB)」であれば，イベントAが起こり，かつイベントBも起こっているときのイベントという意味になる。

＊7　OrEventメソッド
OrEventメソッドは，「AまたはB」でイベント（DAEventオブジェクト）どうしの論理和を求める。「Set newEv = m.OrEvent(eventA, eventB)」であれば，イベントAが起こるか，またはイベントBが起こったときのイベントという意味になる。

使用イメージ①



《第5図》これは[グー][チョキ][パー]のGIFファイル

使用イメージ②

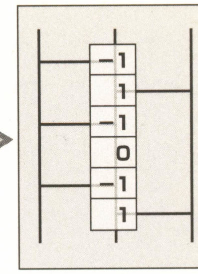


《第6図》[勝ち][負け][引きわけ]のGIFファイル

あみだくじのパターン

値	値の意味
−1	← 中央から左へ
1	→ 中央から右へ
−1	← 中央から左へ
0	(線なし)
−1	← 中央から左へ
1	→ 中央から右へ

あみだ配列



《第7図》パターン配列はこのようになっている

と“ジャンケン”を組み合わせた単純なゲームです。ルールもジャンケンと同じです。

サンプルを実行すると，**第4図①**のような画面が表示されます。上部の[グー][チョキ][パー]のいずれかをクリックすると，不要な横線が消えてあみだくじが現れます。選んだものは，**第4図②**のように経路に沿って下へ移動していきます。**第4図③**のように，ゴールインした相手とのジャンケンになります。最後に勝敗の判定と選択した経路が太字で表示されます。

勝敗は以下のどれかになります（**第4図右側**）

- **勝ち(WIN)**
- **負け(LOSS)**
- **引きわけ(DRAW)**

もう一度勝負する場合は，画面左上の[RELOAD]ボタンをクリックしてください。

サンプルで使用しているイメージ・ファイルは，以下のようになります。

- **goo.gif／cho.gif／paa.gif（第5図）**
  [グー][チョキ][パー]のGIFファイルです。
- **WIN.gif／LOS.gif／TIE.gif（第6図）**
  [勝ち][負け][引きわけ]のGIFファイルです。
- **BACK055.GIF**[*8]
  背景として使うGIFファイルです。

*8 BACK055.GIF
このGIFファイル“BACK055.GIF”は，株式会社インプレス『Hybrid版Internetホームページ素材集 Cool＆ビジネス編』から使用しています。データの著作権は株式会社インプレスに帰属します。

プログラム解説 まずはあみだくじのパターン

このプログラムは，一番はじめにあみだくじのパターンを作成します。

あみだくじのパターンは，**第7図**のように，[−1][0][＋1]の値を持つ配列変数（amida）によって決定されます。配列変数（amida）は，実行時に定数BARSの数で初期化され，Random Numberプロシージャによって[−1][0][＋1]のいずれかの値がセットされます。

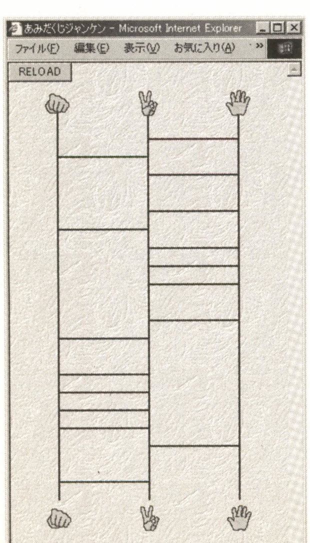
配列の値（−1, 0, ＋1）は，中央の縦線から左右どちらに横線を引くか，または横線を引かないということを意味しています。中央の縦線を基準にすると,「左右同時に横線を引くことはない」というあみだくじの原理をもとに設計しています。

定数BARSで，あみだの横線の本数を変更できます。たとえばこの値を「Const BARS = 20」とすると，**写真4**のように複雑にできます。

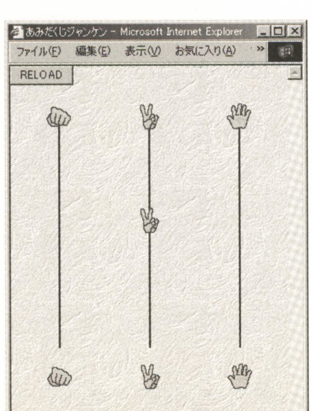
パターンの数は，3のBARS乗とおりです。デフォルトでは「BARS = 6」なので，$3^6=729$とおりとなります。横線を引かない0の出現率を下げていますが，**写真5**のような縦線だけという可能性もあります。

移動経路は点の集合だ！

あみだくじのパターンの配列（amida）をもとに，MakeAmidaArrayプロシージャで，[グー][チョキ][パー]がそれぞれ通過する点（x, y）を

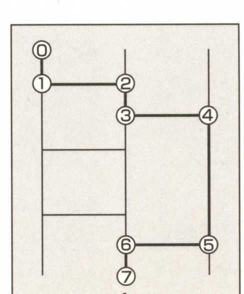


《写真4》BARSを20に変更した状態



《写真5》こんなあみだくじパターンもありうるのだ

あみだくじの移動



[グー]の経路

添字	座標配列のデータ
0	m.Point2(−100, −128)
1	m.Point2(−100, −96)
2	m.Point2(0, −96)
3	m.Point2(0, −64)
4	m.Point2(100, −64)
5	m.Point2(100, 64)
6	m.Point2(0, 64)
7	m.Point2(0, 96)

座標配列

《第8図》[グー]の経路と座標配列のデータ。よく見くらべてみよう

算出して，DAPoint2オブジェクト*9として座標配列の変数goo，cho，paaに格納します。

第8図は，[グー]の座標配列の中身を示しています。

はじまる前に移動経路はきまっている

[グー][チョキ][パー]のそれぞれの座標配列をもとにPolylineメソッドで軌跡を定義して，**第9図**のように [グー][チョキ][パー]それぞれの経路イメージを合成しておきます。

また，定義した軌跡（pthA, pthB, pthC）は，FollowPathAnimメソッドに渡して，[グー][チョキ][パー]の移動に使います。

クリックすると移動がはじまる！

上部の[グー][チョキ][パー]のイメージは，ピッカブルとしてそれぞれ独立しています。

クリックすると，軌跡に沿って移動をはじめます。なお，このとき，クリックされなかった残りの二つのイメージが無効になるように，わざと画面外（Y座標＝－999）へ移動するように仕込んであります。

勝敗判定とイメージの表示

ゴールすると，勝敗イメージと選択経路（太線）イメージを表示します。

あみだくじのパターンが作成された段階で勝敗が決定しているためあらかじめ[グー][チョキ][パー]のそれぞれの勝敗イメージを準備しておきます。勝敗は，それぞれの座標配列データをもとにJudgementプロシージャで判断します。

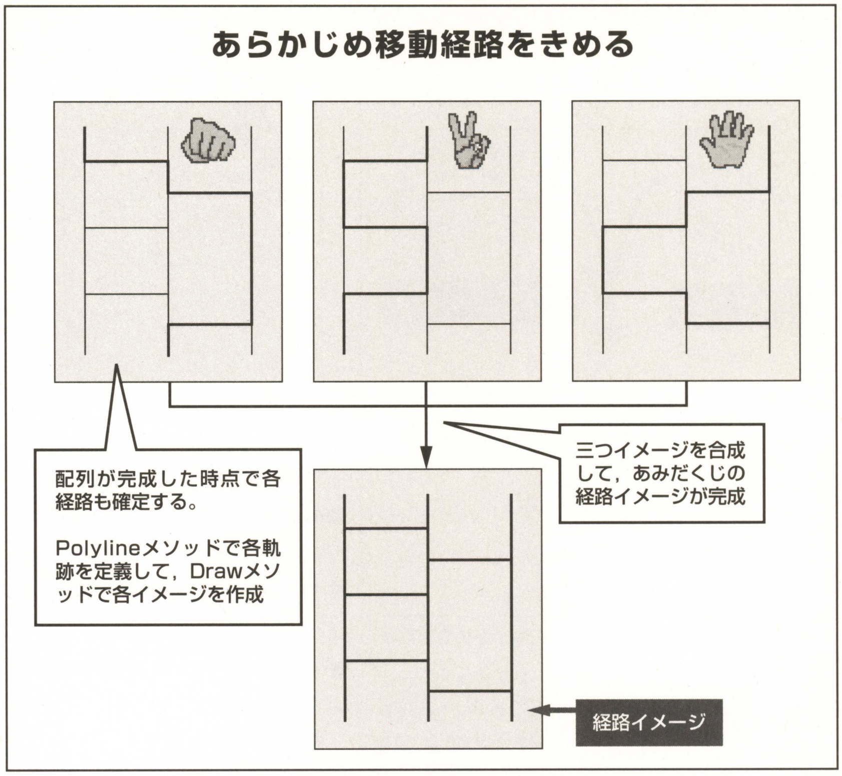
全体の勝敗パターンは，つぎのどれかになります。

- **[グー][チョキ][パー]のどれを選んでも“勝ち”**
- **[グー][チョキ][パー]のどれを選んでも“負け”**
- **[グー][チョキ][パー]のどれを選んでも“引きわけ”**
- **選ぶもので，“勝ち”“負け”“引きわけ”のどれかになる**

勝敗＆選択経路イメージの表示は，イメージの透明度を表すOpacityAnimメソッドと変動値を用いて実現しています。はじめは，0で透明（見えない）にしておき，ゴールした時点で1に変更して不透明（見える）にします。

ゴールしたときのイベントは，つぎのコードのように「はじめは0，グーをクリック（evA）したら，タイマーを起動してPLAY_TIME秒経過したら1にする」という，ピッカブル・イベントとタイマーイベントを組み合わせています。

```
Set ev = m.Timer(PLAY_TIME)
Set num = m.Until(zero, evA,
m.Until(zero, ev, one))
```



《第9図》[グー][チョキ][パー]の経路イメージはこうなっている

スタート時に経路を隠す

実際のあみだくじでは，はじめにあみだの部分を隠しておきます。

このプログラムでは，**第10図**のようにあみだのイメージに横線を引いたグリッド・イメージを合成しておき，本来の経路を見破られないようにしています。グリッドイメージは，[グー][チョキ][パー]のいずれかをクリックすると，**第11図**のように縮小して消えるように設定してあります。

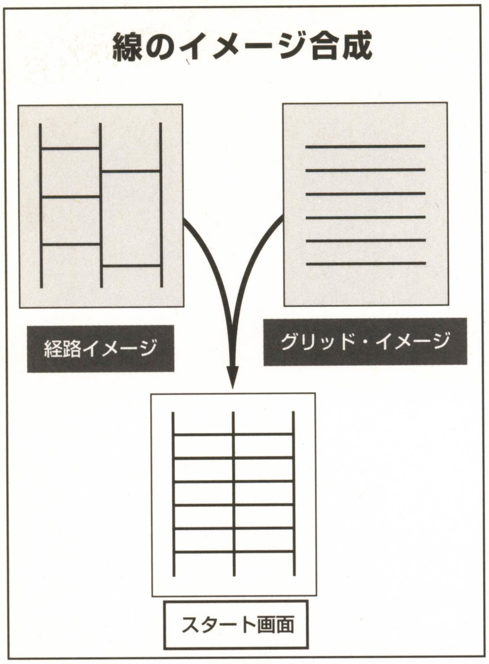
使用しているイメージ

このプログラムは，つぎのようないくつものイメージ（DAImageオブジェクト）を合成しています（**第12図**）。

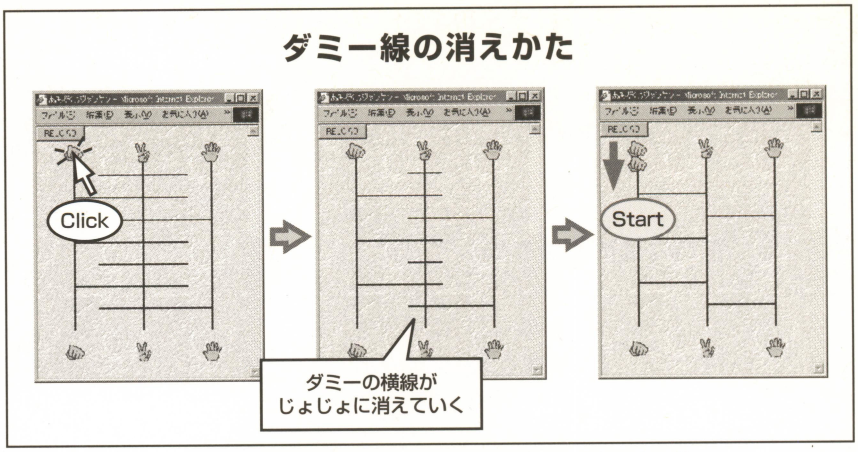
●グー・チョキ・パーのイメージ（icons）

[グー][チョキ][パー]の各ピッカブルからなるイメージで，クリックによりそれぞれ軌跡に沿

＊9　DAPoint2オブジェクト
DirectAnimationでは，2次元（平面）座標上の点（x，y）をDAPoint2オブジェクトで管理している。Point2メソッドを使って，X，Yの座標値からDAPoint2オブジェクトを作成できる。たとえば点（4，5）は，「m.Point2(4, 5)」という式となる。逆に，DAPoint2オブジェクトのXプロパティ，Yプロパティを参照すれば，それぞれX座標，Y座標の値がDANumber型として取得できる。数値として取得する場合は，「X.Extract」や「Y.Extract」のようにExtractメソッドを使う。



《第10図》はじめにあみだくじの経路を隠しておくのだ



《第11図》不要な横線は消えて，あみだくじが見えてくるぞ

リスト1〜リスト4
入力&実行方法

このリストは，IE5／5.5とテキスト・エディタ（『メモ帳』など）があれば，実行することが可能だ。IE5は，以下のアドレスから無償でダウンロードすることができる。

●http://www.microsoft.com/windows/ie_intl/ja/download/

リストをメモ帳などで入力したら，拡張子を"htm"または"html"にして保存する。そして，IE5の[ファイル]メニューより，[開く]で保存したファイルを読み込もう。すると，リストが実行されるようすを，IE5で確認することができる。

って移動します。

●パネル・イメージ(panel)

上部と下部の[グー][チョキ][パー]を表示するためのイメージです。

●グリッド・イメージ(grid)

スタート時にあみだ（経路イメージ）をごまかすためにかぶせる，横線のイメージです。[グー][チョキ][パー]のいずれかをクリックすると，縮小して見えなくなります。

●勝敗&選択経路イメージ(grid)

ゴールしたときに，表示する勝敗イメージ（WIN，LOSS，DRAW）と，選択経路を太線で表したイメージを合成したイメージです。はじめは透明で，ゴールすると不透明になり画面上に現れます。

●経路イメージ(lines)

[グー][チョキ][パー]のそれぞれの経路を合成したイメージです。

●背景イメージ(back)

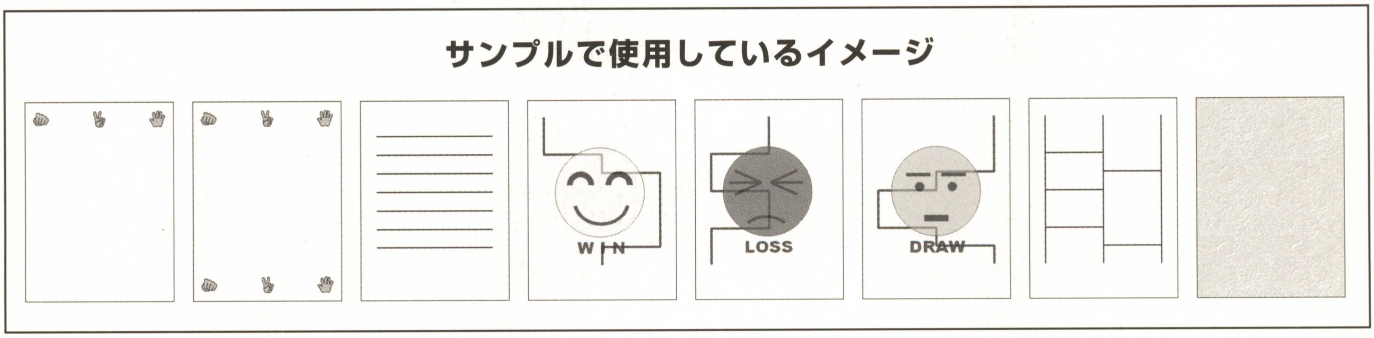
バックグランド用の，タイリングされた画像イメージです。

まとめ

今回は，プログラム作成したのがちょうど夏休みだったので，ゲームっぽい動きをするアニメーションを作ってみました。

『Microsoft Visual Basic（マイクロソフト ビジュアル ベーシック）』や『Microsoft Visual C＋＋（マイクロソフト ビジュアル シープラスプラス）』を使ったDirectXのゲームにははるかにおよびませんが，DirectAnimationでもここまで作れるのだという一つの例になったと思います。

みなさんも，ぜひゲーム制作に挑戦してみてください（次回もゲームを作ってみようかなぁっと）。



《第12図》『あみだくじジャンケン』では，これらのDAImageオブジェクトを使用している

《リスト1》FollowPathを使ったサンプル『くるくる天使』"Angel.htm"（VBScript用）

```
<HTML>
<TITLE>くるくるエンジェル</TITLE>
<BODY LEFTMARGIN = "0", TOPMARGIN = "0">
<SCRIPT LANGUAGE = "VBScript">
'//------------------------------------------------------
'// DirectAnimation Sample                    by Masato Hirai
'//
'//   『くるくるエンジェル』 Sample of FollowPath
'//------------------------------------------------------
Option Explicit

'//------------------------------------------------------
'// 初期化処理
'//------------------------------------------------------
Sub Window_onLoad()
  '// 変数・定数の宣言
  Dim m       'As DAStatics       'DAStaticsオブジェクト
  Dim pth     'As DAPath2         '軌跡
  Dim tran    'As DATransform2    '移動
  Dim img     'As DAImage         '画像イメージ
  Dim lines   'As DAImage         '軌跡イメージ
  '// DirectAnimationの初期化（ピクセル座標系）
  Set m = DAControl.PixelLibrary
  '// 画像ファイルのロード
  Set img = m.ImportImage("CH210.GIF")
  '// 軌跡の定義
  Set pth = m.Oval(300, 200)    '楕円
  '// 軌跡に沿った移動の定義
  Set tran = m.FollowPath(pth, 5).RepeatForever
  '// 移動を画像イメージに適用
  Set img = img.Transform(tran)
  '// 軌跡イメージ(経路を表示するための処理)
  Set lines = pth.Draw(m.DefaultLineStyle)
  '// コントロールにオブジェクトをセットしてスタート
  DAControl.Image = m.Overlay(img, lines)
  DAControl.Start
  Call Window_onResize()
End Sub

'--------------------------------------------------------
'ウィンドウのサイズ変更時の処理
'--------------------------------------------------------
Sub Window_onResize()
  On Error Resume Next
  DAControl.Width  = Document.Body.ClientWidth
  DAControl.Height = Document.Body.ClientHeight
End Sub

</SCRIPT>
<! DirectAnimation コントロール>
<OBJECT ID = "DAControl"
  CLASSID = "CLSID:B6FFC24C-7E13-11D0-9B47-00C04FC2F51D"
  STYLE = "position:absolute; left:0; top:0; z-index:-1">
</OBJECT>
</BODY>
</HTML>
```

《リスト2》FollowPathAngleを使ったサンプル『てんとう虫』“Ladybug.htm”（VBScript用）

```
<HTML>
<TITLE>てんとう虫</TITLE>
<BODY LEFTMARGIN = "0", TOPMARGIN = "0">
<SCRIPT LANGUAGE = "VBScript">
'//-----------------------------------------------------------
'// DirectAnimation Sample                       by Masato Hirai
'//
'//    『てんとう虫』 Sample of FollowPathAngle
'//-----------------------------------------------------------
Option Explicit

'//-----------------------------------------------------------
'// 初期化処理
'//-----------------------------------------------------------
Sub Window_onLoad()
  '// 変数・定数の宣言
  Dim m      'As DAStatics      'DAStaticsオブジェクト
  Dim pth    'As DAPath2        '軌跡
  Dim tran   'As DATransform2   '移動
  Dim img    'As DAImage        '画像イメージ
  Dim lines  'As DAImage        '軌跡イメージ
  '// DirectAnimationの初期化（ピクセル座標系）
  Set m = DAControl.PixelLibrary
  '// 画像ファイルのロード
  Set img = m.ImportImage("TENTO.GIF")
  '// 軌跡の定義
  Dim xy(9)   '星型
  Set xy(0) = m.Point2(0, 140)
  Set xy(1) = m.Point2(-36, 32)
  Set xy(2) = m.Point2(-152, 32)
  Set xy(3) = m.Point2(-56, -36)
  Set xy(4) = m.Point2(-96, -144)
  Set xy(5) = m.Point2(0, -76)
  Set xy(6) = m.Point2(96, -144)
  Set xy(7) = m.Point2(56, -36)
  Set xy(8) = m.Point2(152, 32)
  Set xy(9) = m.Point2(36, 32)
  Set pth = m.Polyline(xy).Close
  '// 軌跡に沿った移動の定義
  Set tran = m.FollowPathAngle(pth, 5).RepeatForever
  '// 移動を画像イメージに適用
  Set img = img.Transform(tran)
  '// 軌跡イメージ(経路を表示するための処理)
  Set lines = pth.Draw(m.DefaultLineStyle)
  '// コントロールにオブジェクトをセットしてスタート
  DAControl.Image = m.Overlay(img, lines)
  DAControl.Start
  Call Window_onResize()
End Sub

'------------------------------------------------------------
'ウィンドウのサイズ変更時の処理
'------------------------------------------------------------
Sub Window_onResize()
  On Error Resume Next
  DAControl.Width  = Document.Body.ClientWidth
  DAControl.Height = Document.Body.ClientHeight
End Sub

</SCRIPT>
<! DirectAnimation コントロール>
<OBJECT ID = "DAControl"
  CLASSID = "CLSID:B6FFC24C-7E13-11D0-9B47-00C04FC2F51D"
  STYLE = "position:absolute; left:0; top:0; z-index:-1">
</OBJECT>
</BODY>
</HTML>
```

《リスト3》サンプル『ピッカブル・イベント』“Pickable.htm”（VBScript用）

```
<HTML>
<TITLE>ピッカブル・イベント</TITLE>
<BODY LEFTMARGIN = "0", TOPMARGIN = "0">
<SCRIPT LANGUAGE = "VBScript">
'//-----------------------------------------------------------
'// DirectAnimation Sample                       by Masato Hirai
'//
'//    『ピッカブル・イベント』 Sample of Pickable
'//-----------------------------------------------------------
Option Explicit

'//-----------------------------------------------------------
'// 初期化処理
'//-----------------------------------------------------------
Sub Window_onLoad()
  '//
  '// 変数・定数の宣言
  '//
  Dim m      'As DAStatics        'DAStaticsオブジェクト
  Dim panel  'As DAImage          '塗りつぶし用のイメージ
  Dim border 'As DALineStyle      '枠線のスタイル
  Dim rate   'As DANumber         '回転／停止を制御する変動値
  Dim img    'As DAImage          '扇形イメージ
  Dim pic    'As DAPickableResult 'ピッカブル
  Dim ev     'As DAEvent          'イベント
  '//
  '// DirectAnimationの初期化（ピクセル座標系）
  '//
  Set m = DAControl.PixelLibrary
  '//
  '// 扇形イメージ
  '//
  Set border = m.DefaultLineStyle.Width(4)
  Set panel = m.SolidColorImage(m.Yellow)
  Set img = m.PieDegrees(30, 330, 200, 200).Fill(border, panel)
  Set img = img.Transform(m.Rotate2RateDegrees(360))
  '//
  '// ピッカブルの作成
  '//
  Set pic = img.Pickable
  '//
  '// ピッカブル・イベントの取得
  '//
  Set ev = pic.PickEvent
  '//
  '// イベントの組み合わせ
  '//
  Set ev = m.AndEvent(m.LeftButtonDown, ev)
  '//
  '// ピッカブルからイメージを取得
  '//
  Set img = pic.Image
  '//
  '// イベントを使用して変動値を定義
  '//
  Set rate = CreateObject("DirectAnimation.DANumber")
  rate.Init m.Until(m.DANumber(1), ev, _
            m.Until(m.DANumber(0), ev, rate))
  Set img = img.SubstituteTime(m.Integral(rate))
  '//
  '// コントロールにオブジェクトをセットしてスタート
  '//
  DAControl.Image = img
  DAControl.Start
  Call Window_onResize()
End Sub

'//-----------------------------------------------------------
'// ウィンドウのサイズ変更時の処理
'//-----------------------------------------------------------
Sub Window_onResize()
  On Error Resume Next
  DAControl.Width  = Document.Body.ClientWidth
  DAControl.Height = Document.Body.ClientHeight
End Sub

</SCRIPT>
<! DirectAnimation コントロール>
<OBJECT ID = "DAControl"
  CLASSID = "CLSID:B6FFC24C-7E13-11D0-9B47-00C04FC2F51D"
  STYLE = "position:absolute; left:0; top:0; z-index:-1">
</OBJECT>
</BODY>
</HTML>
```

《リスト4》サンプル『あみだくじジャンケン』“Janken.htm”（VBScript用）

```
<HTML>
<TITLE>あみだくじジャンケン</TITLE>
<BODY LEFTMARGIN = "0", TOPMARGIN = "0">
<SCRIPT LANGUAGE = "VBScript">
'//-----------------------------------------------------------
'//
'// DirectAnimation Sample                       by Masato Hirai
'//
'//    『あみだくじジャンケン』
'//
'//-----------------------------------------------------------
Option Explicit
'//
'// パブリック変数・定数の宣言
'//
Const WW = 100                    'あみだの縦線の(左右)間隔
Const HH = 32                     'あみだの横線の(上下)間隔
Const BARS = 6                    'あみだの横線の本数
Const SIZE = 2                    'あみだの線の太さ
Const PLAY_TIME = 5               'ジャンケンイメージの移動時間
Const WIN = 1                     '勝敗判定フラグ：勝ち
Const TIE = 0                     '勝敗判定フラグ：引き分け
Const LOS = -1                    '勝敗判定フラグ：負け
Dim m       'As DAStatics         'DAStaticsオブジェクト
Dim amida() 'As Variant           'あみだくじのパターンの配列
Dim ADJ     'As Double            '上下位置の調整用

'//-----------------------------------------------------------
'// 初期化処理
'//-----------------------------------------------------------
Sub Window_onLoad()
  '//
  '// 変数・定数の宣言
  '//
  Dim icoA   'As DAImage          '[グー]イメージ
  Dim icoB   'As DAImage          '[チョキ]イメージ
  Dim icoC   'As DAImage          '[パー]イメージ
  Dim pthA   'As DAPath2          '[グー]の軌跡(移動経路)
  Dim pthB   'As DAPath2          '[チョキ]の軌跡(移動経路)
  Dim pthC   'As DAPath2          '[パー]の軌跡(移動経路)
  Dim evA    'As DAEvent          '[グー]のクリックイベント
  Dim evB    'As DAEvent          '[チョキ]のクリックイベント
  Dim evC    'As DAEvent          '[パー]のクリックイベント
  Dim icons  'As DAImage          'グーチョキパーイメージ
  Dim panel  'As DAImage          'パネルイメージ
  Dim lines  'As DAImage          '経路イメージ
  Dim result 'As DAImage          '勝敗＆選択経路イメージ
  Dim grid   'As DAImage          'グリッドイメージ
  Dim back   'As DAImage          '背景イメージ
  Dim border 'As DALineStyle      '線種
  Dim tmp    'As DAImage          '作業用
  Dim tran   'As DATransform2     '作業用
  Dim num    'As DANumber         '作業用
  Dim pk     'As DAPickableResult '作業用
  Dim ev     'As DAEvent          '作業用
  Dim zero   'As DANumber         'm.DANumber(0)用
  Dim one    'As DANumber         'm.DANumber(1)用
  Dim final  'As Variant          '各イメージのバリアント配列
  Dim k      'As Integer          'ループカウンタ
  '//
  '// DirectAnimationの初期化（ピクセル座標系）
  '//
  Set m = DAControl.PixelLibrary
  Set zero = m.DANumber(0)        '← m.DANumber(0)
  Set one = m.DANumber(1)         '← m.DANumber(1)
  ADJ = ((BARS + 1) * HH / 2)     '← 全高さの1/2
  '//
  '// あみだくじのパターンの準備
  '//
  Randomize                       '← 乱数ジェネレータの初期化
  ReDim amida(BARS - 1)           '← 指定された本数分の配列
  For k = 0 To UBound(amida)
    amida(k) = RandomNumber()     '← ランダムに-1, 0, +1を設定
  Next
  '//
  '// 移動経路の座標配列の作成
  '//
  ReDim paa(0) 'As DAPoint2       '[グー]の移動経路の座標配列
  ReDim goo(0) 'As DAPoint2       '[チョキ]の移動経路の座標配列
  ReDim cho(0) 'As DAPoint2       '[パー]の移動経路の座標配列
  Call MakeAmidaArray(goo, -1)
  Call MakeAmidaArray(cho, 0)
  Call MakeAmidaArray(paa, 1)
  '//
  '// [グー].[チョキ].[パー]のGIFファイルをロード
  '//
  Set icoA = m.ImportImage("goo.gif")  '[グー]
  Set icoB = m.ImportImage("cho.gif")  '[チョキ]
  Set icoC = m.ImportImage("paa.gif")  '[パー]
  '//
  '// パネルイメージ（画面上に表示するだけ）
  '//
  Set panel = m.EmptyImage
  '上側の[グー].[チョキ].[パー]
  Set tran = m.Translate2(-WW, ADJ + HH)
  Set panel = m.Overlay(icoA.Transform(tran), panel)
  Set tran = m.Translate2(0, ADJ + HH)
  Set panel = m.Overlay(icoB.Transform(tran), panel)
  Set tran = m.Translate2(WW, ADJ + HH)
  Set panel = m.Overlay(icoC.Transform(tran), panel)
  '下側の[グー].[チョキ].[パー]
  Set tran = m.Translate2(-WW, -ADJ - HH)
  Set panel = m.Overlay(icoA.Transform(tran), panel)
  Set tran = m.Translate2(0, -ADJ - HH)
  Set panel = m.Overlay(icoB.Transform(tran), panel)
  Set tran = m.Translate2(WW, -ADJ - HH)
  Set panel = m.Overlay(icoC.Transform(tran), panel)
  '//
  '// 経路イメージ（あみだくじ）
  '//
  Set pthA = m.Polyline(goo)      '[グー]の軌跡
  Set pthB = m.Polyline(cho)      '[チョキ]の軌跡
  Set pthC = m.Polyline(paa)      '[パー]の軌跡
  Set border = m.DefaultLineStyle.Width(SIZE)
  Set lines = m.EmptyImage
  Set lines = m.Overlay(pthA.Draw(border), lines)
  Set lines = m.Overlay(pthB.Draw(border), lines)
  Set lines = m.Overlay(pthC.Draw(border), lines)
  '//
  '// グーチョキパーイメージ
  '//
  '[グー]イメージにイベント処理と移動を付加
  Set pk = icoA.Pickable
  Set evA = m.AndEvent(m.LeftButtonDown, pk.PickEvent)
  Set icoA = pk.Image
  Set num = m.Until(zero, evA, m.Interpolate(0, 1, PLAY_TIME))
  Set icoA = icoA.Transform(m.FollowPathAnim(pthA, num))
  '[チョキ]イメージにイベント処理と移動を付加
  Set pk = icoB.Pickable
  Set evB = m.AndEvent(m.LeftButtonDown, pk.PickEvent)
  Set icoB = pk.Image
  Set num = m.Until(zero, evB, m.Interpolate(0, 1, PLAY_TIME))
  Set icoB = icoB.Transform(m.FollowPathAnim(pthB, num))
  '[パー]イメージにイベント処理と移動を付加
  Set pk = icoC.Pickable
  Set evC = m.AndEvent(m.LeftButtonDown, pk.PickEvent)
  Set icoC = pk.Image
  Set num = m.Until(zero, evC, m.Interpolate(0, 1, PLAY_TIME))
  Set icoC = icoC.Transform(m.FollowPathAnim(pthC, num))
  'いづれかがクリックされたら残りを無効にする
  Set num = m.Until(zero, m.OrEvent(evB, evC), m.DANumber(-999))
  Set icoA = icoA.Transform(m.Translate2Anim(zero, num))
  Set num = m.Until(zero, m.OrEvent(evA, evC), m.DANumber(-999))
  Set icoB = icoB.Transform(m.Translate2Anim(zero, num))
  Set num = m.Until(zero, m.OrEvent(evA, evB), m.DANumber(-999))
  Set icoC = icoC.Transform(m.Translate2Anim(zero, num))
  '3つを1つのイメージにまとめる
  Set icons = m.OverlayArray(Array(icoA, icoB, icoC))
  '//
  '// 勝敗＆選択経路イメージ（移動完了時に勝敗と選択経路を表示）
  '//
  Set ev = m.Timer(PLAY_TIME)     '← タイマーイベントを定義
  Set result = m.EmptyImage
  Set border = m.DefaultLineStyle.Color(m.Blue).Width(SIZE * 2)
  '[グー]の場合の勝敗(画像ファイル)と選択経路(太目の線)
  Set tmp = m.Overlay(MakeJudgeImage(goo), pthA.Draw(border))
  Set num = m.Until(zero, evA, m.Until(zero, ev, one))
  Set result = m.Overlay(tmp.OpacityAnim(num), result)
  '[チョキ]の場合の勝敗(画像ファイル)と選択経路(太目の線)
  Set tmp = m.Overlay(MakeJudgeImage(cho), pthB.Draw(border))
  Set num = m.Until(zero, evB, m.Until(zero, ev, one))
  Set result = m.Overlay(tmp.OpacityAnim(num), result)
  '[パー]の場合の勝敗(画像ファイル)と選択経路(太目の線)
  Set tmp = m.Overlay(MakeJudgeImage(paa), pthC.Draw(border))
  Set num = m.Until(zero, evC, m.Until(zero, ev, one))
  Set result = m.Overlay(tmp.OpacityAnim(num), result)
  '//
  '// グリッドイメージ（横線で本来の経路を隠す）
  '//
  Set grid = MakeGridImage()
  '3つのうちいづれかがクリックされたら縮小して消す
  Set ev = m.OrEvent(evA, m.OrEvent(evB, evC))
  Set num = m.Until(one, ev, m.Interpolate(1, 0, 0.5))
  Set grid = grid.Transform(m.Scale2Anim(num, one))
  '//
  '// 背景イメージ（タイリング）
  '//
  Set back = m.ImportImage("BACK055.GIF").Tile
  '//
  '// コントロールにオブジェクトをセットしてスタート
  '//
  final = Array(icons, panel, grid, result, lines, back)
  DAControl.Image = m.OverlayArray(final)
  DAControl.Start
  Call Window_onResize()
End Sub

'//-----------------------------------------------------------
'// ウィンドウのサイズ変更時の処理
'//-----------------------------------------------------------
Sub Window_onResize()
  On Error Resume Next
  DAControl.Width  = Document.Body.ClientWidth
  DAControl.Height = Document.Body.ClientHeight
End Sub

'//-----------------------------------------------------------
'// ランダムに [-1], [0], [+1]を返す
'//-----------------------------------------------------------
Function RandomNumber()
  Dim d      'As Double           '乱数
  d = Rnd() * 3                   '← 0.0～3.0の範囲での乱数
  If 0.25 < d And d <= 1.5 Then
    RandomNumber = -1
  ElseIf 1.5 < d And d <= 2.75 Then
    RandomNumber = 1
  Else                            '[0]の出現率は低い
    RandomNumber = 0
  End If
End Function

'//-----------------------------------------------------------
'// [グー],[チョキ],[パー]の移動経路の座標配列を作成
'//-----------------------------------------------------------
Sub MakeAmidaArray(arr(), Byval xpos)
  Dim n      'As Long             'ループカウンタ
  Dim ypos   'As Double           'Y座標
  Dim img    'As DAImage          '作業用イメージ
  'スタート地点
  ReDim arr(0)
  Set arr(0) = m.Point2(xpos * WW, -HH - ADJ)
  For n = 0 To UBound(amida)
    ypos = (n + 1) * HH - ADJ
    If xpos = -1 And amida(n) = -1 Then
      '[|→|  |]の場合（左から中央）
      ReDim Preserve arr(UBound(arr) + 2)
      Set arr(UBound(arr) - 1) = m.Point2(xpos * WW, ypos)
      xpos = 0
      Set arr(UBound(arr)) = m.Point2(xpos * WW, ypos)
    ElseIf xpos = 0 And amida(n) = 1 Then
      '[|  |→|]の場合（中央から右）
      ReDim Preserve arr(UBound(arr) + 2)
      Set arr(UBound(arr) - 1) = m.Point2(xpos * WW, ypos)
      xpos = 1
      Set arr(UBound(arr)) = m.Point2(xpos * WW, ypos)
    ElseIf xpos = 0 And amida(n) = -1 Then
      '[|←|  |]の場合（中央から左）
      ReDim Preserve arr(UBound(arr) + 2)
      Set arr(UBound(arr) - 1) = m.Point2(xpos * WW, ypos)
      xpos = -1
      Set arr(UBound(arr)) = m.Point2(xpos * WW, ypos)
    ElseIf xpos = 1 And amida(n) = 1 Then
      '[|  |←|]の場合（右から中央）
      ReDim Preserve arr(UBound(arr) + 2)
      Set arr(UBound(arr) - 1) = m.Point2(xpos * WW, ypos)
      xpos = 0
      Set arr(UBound(arr)) = m.Point2(xpos * WW, ypos)
    End If
  Next
  'ゴール地点
  ypos = (n + 1) * HH - ADJ
  ReDim Preserve arr(UBound(arr) + 1)
  Set arr(UBound(arr)) = m.Point2(xpos * WW, ypos)
End Sub

'//-----------------------------------------------------------
'// グリッドイメージの作成
'//-----------------------------------------------------------
Function MakeGridImage()
  Dim k      'As Integer          'ループカウンタ
  Dim img    'As DAImage          '作業用イメージ
  Dim pth    'As DAPath2          '横線の軌跡
  Dim border 'As DALineStyle      '線種
  Set border = m.DefaultLineStyle.Width(SIZE)
  Set img = m.EmptyImage
  For k = 0 To BARS
    Set pth = m.Line(m.Point2(-WW, k * HH - ADJ), _
                     m.Point2(WW, k * HH - ADJ))
    Set img = m.Overlay(pth.Draw(border), img)
  Next
  Set MakeGridImage = img
End Function

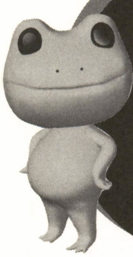
'//-----------------------------------------------------------
'// 勝敗イメージの作成
'//-----------------------------------------------------------
Function MakeJudgeImage(arr())
  Dim img    'As DAImage          '作業用イメージ
  Select Case Judgement(arr)      '← 勝敗判定プロシージャ
    Case WIN  '[勝ち]の場合 "win.gif"を半透明で表示
      Set img = m.ImportImage("win.gif").Opacity(0.7)
    Case TIE  '[引き分け]の場合 "tie.gif"を半透明で表示
      Set img = m.ImportImage("tie.gif").Opacity(0.7)
    Case LOS  '[負け]の場合 "los.gif"を半透明で表示
      Set img = m.ImportImage("los.gif").Opacity(0.7)
  End Select
  Set MakeJudgeImage = img
End Function

'//-----------------------------------------------------------
'// 座標配列から勝敗を判定してフラグ(WIN, LOS, TIE)を返す
'//-----------------------------------------------------------
Function Judgement(arr())
  '配列の先頭X位置と末尾X位置か決定
  If arr(0).x.Extract < 0 Then        '[グー]を選択した場合
    If arr(UBound(arr)).x.Extract < 0 Then
      Judgement = TIE   '引き分け
    ElseIf arr(UBound(arr)).x.Extract = 0 Then
      Judgement = WIN   '勝ち
    Else
      Judgement = LOS   '負け
    End If
  ElseIf arr(0).x.Extract = 0 Then  '[チョキ]を選択した場合
    If arr(UBound(arr)).x.Extract < 0 Then
      Judgement = LOS   '負け
    ElseIf arr(UBound(arr)).x.Extract = 0 Then
      Judgement = TIE   '引き分け
    Else
      Judgement = WIN   '勝ち
    End If
  Else                              '[パー]を選択した場合
    If arr(UBound(arr)).x.Extract < 0 Then
      Judgement = WIN   '負け
    ElseIf arr(UBound(arr)).x.Extract = 0 Then
      Judgement = LOS   '引き分け
    Else
      Judgement = TIE   '勝ち
    End If
  End If
End Function

'//-----------------------------------------------------------
'// [RELOAD]ボタン
'//-----------------------------------------------------------
Sub button1_onclick
  window.navigate("Janken.htm")
End Sub

</SCRIPT>
<! DirectAnimation コントロール>
<OBJECT ID = "DAControl"
  CLASSID = "CLSID:B6FFC24C-7E13-11D0-9B47-00C04FC2F51D"
  STYLE = "position:absolute; left:0; top:0; z-index:-1">
</OBJECT>
<! RELOAD ボタン>
<INPUT ID="Button1" type=button value="RELOAD" name=Button1>

</BODY>
</HTML>
```



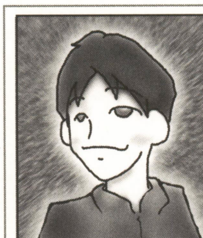
プログラマーをめざす人の必見講座

図解! Visual Basic ゲーム作成術

第8回　キャラクター・アクション・ゲーム②方向の数値化

キャラクターの移動方向が“上下左右”の4とおりになるキャラクター・アクション・ゲーム。この“方向”を条件判断文などによる場合わけで処理すると，どうしても処理が4とおりに分岐することになるため，プログラムが長くなってしまいます。これに対し，テーブル処理によってうまく処理すれば，プログラムの簡略化と高速化が可能になります。今回は，方向をプログラムで数値化して取り扱う方法と，それによって実現が容易になるなめらかな移動アルゴリズムを，『Microsoft Visual Basic（以下VBと略）』で作成してみましょう。

TEXT&PROGRAMMING●東　幸太



BASIC歴15年以上のプログラマー。風の強い台風が近づくときには，停電が気になる。ディスク・アクセス中の停電は，下手をするとドライブのデータを大きく破壊しかねない。UPS（無停電電源）を使えば対応はできるものの，やはりこういう場面ではノート・パソコンが便利だ。

Microsoft Visual Basic Programming System for Windows Ver.6.0（日本語版）

オープン価格

※編集部の価格調査によると，下記価格が目安になります。

ラーニング版2万円強

（アカデミック・パッケージ1万円弱）

プロフェッショナル版8～9万円

（アカデミック・パッケージ1万円強）

問い合わせ先

マイクロソフト

☎03-5454-2300

ホームページ

http://www.microsoft.com/japan/

動作環境

Windows 95／98／2000／NT4.0が動作するマシン

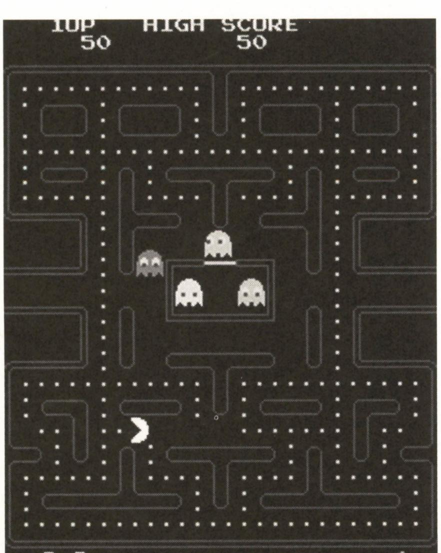


方向を数値で表現する？

『パックマン』などに代表されるキャラクター・アクション・ゲームは，キャラクターが上下左右の4方向に移動しますね（**写真1**）。この移動処理を，**第1図**のような場合わけで処理すると，処理が4とおりずつ必要となってしまい，プログラムが長くなってしまいます。

そこで，「方向を数値化する」ということを考えてみましょう。数値化とは，“上”や“右”といった概念に，数値を対応させることです。概念を直接プログラムに組み込もうとすると，どうしても「IF文」による条件判断が必要となり，結果としてプログラムが**第1図**のような形態になっていきます。これに対し，方向を数値化してやると，IF文以外による条件判断が可能となるため，条件に応じた処理を適用できます。

ここでは，**第2図**のように上下左右の方向に数値を対応させます。これによって，1という数値が“右”を，2という数値が“下”をそれぞれ示すことになります。



《写真1》キャラクター・アクション・ゲームの名作，ナムコ『パックマン』（©namco）

テーブル処理をマスターしよう！

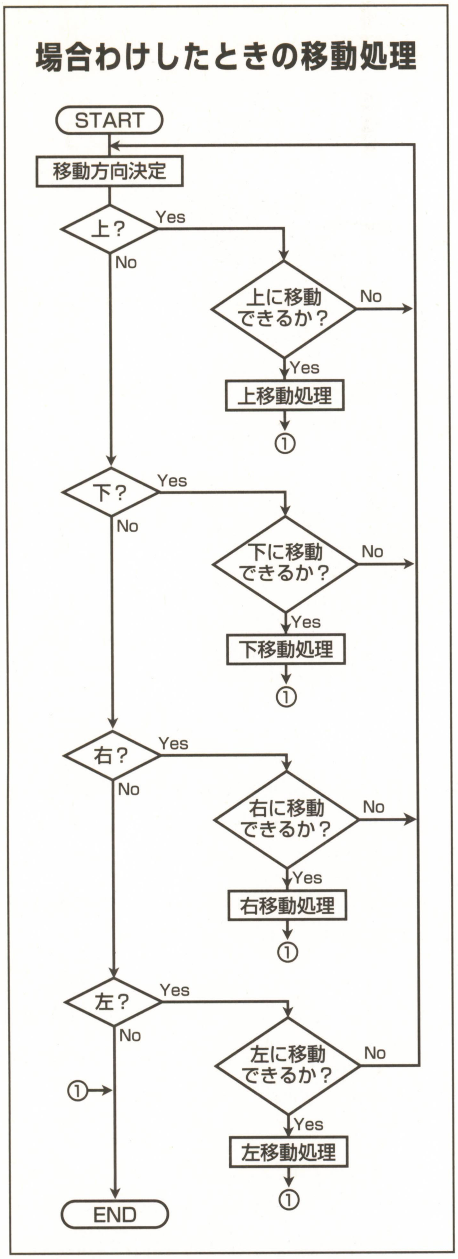
方向を数値化させることそのものには，とくに大きなメリットはありません。しかし，方向が数値化されたことによって，“テーブル処理”が適用できるようになります。

テーブル処理というのは，何らかの数値を目次（インデックス）に見立てて，対応するページに必要なデータなどをあらかじめ用意（テーブル）しておき，必要なデータを目次の数値だけで即座に参照できるようにする処理です。構成としては，**第3図**のようになります。

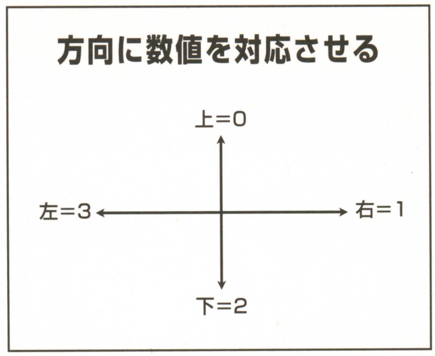
インデックスとなる数値から，必要なデータが簡単に計算できる場合は，テーブル処理はあまり有効とはなりません。たとえば，算数の「九九の表」の場合は，テーブルとして用意するよりも，直接計算したほうが簡単です。しかし，必要なデータが複雑で直接計算できないような場合や，インデックスとデータに数学的な関係がない場合は，テーブルを用意するのが有効です。

ここでは，方向から各座標方向の移動量を求める場合を考えてみましょう。テーブル処理を行わない場合は，**第4図**のように方向をIF文で条件判断し，X方向移動量とY方向移動量を求めることになります。

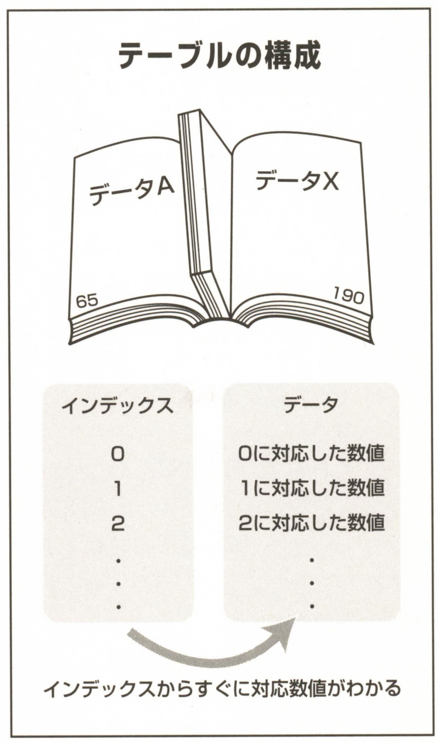
ところが，**第5図**のような移動量配列変数をテーブルとして用意してやると，移動量は配列変数の添え字に数値化され



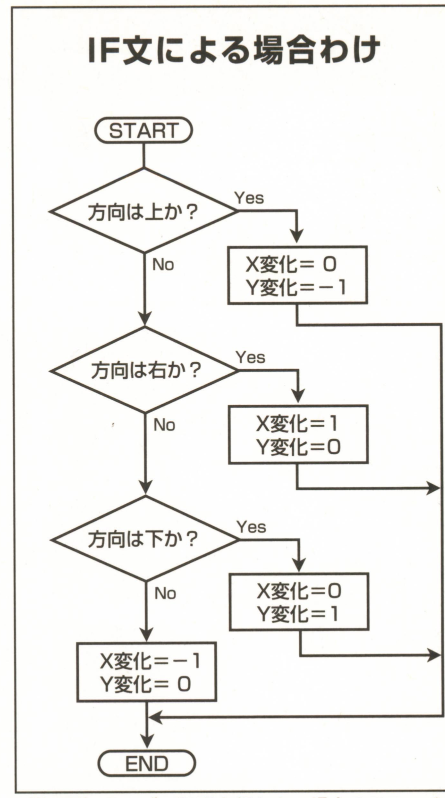
《第1図》このフローチャート（流れ図）を見てもわかるように，プログラムが長くなってしまう



《第2図》上下左右それぞれに，0／1／2／3の数値を対応させるわけだ



《第3図》テーブル処理は，このように「本の目次から対応ページを呼びだして目的の情報を得る」という考えかたに似ている



《第4図》テーブル処理を行わない場合のフローチャートはこうなる。判断が多いのがわかる

テーブルを用いる場合

インデックス	X変化テーブル	Y変化テーブル
0	1	－1
1	0	0
2	－1	1
3	0	0

第2図の方向数値に対応

《第5図》X方向，Y方向に対する移動量をこのようにテーブルとして用意する

た方向をあたえるだけで求められてしまいます。**第4図**の処理と比較すると，その簡単さ，見とおしのよさは圧倒的です。このように，テーブル処理はIF文による条件判断をまとめてしまうことが可能で，そのためには取り扱っている概念を数値化しておくことが必要というわけです。

相対方向の計算をする！

方向を数値化するメリットには，もう一つ大きなものがあります。それは，「相対的な方向を簡単に計算できるようになる」という点です。

キャラクター・アクション・ゲームの場合は，キャラクターに“向き”があります。キャラクターにとっての“右”は，そのキャラクターがどちらを向いているかで変わってきます。キャラクターが上を向いている場合は“右”は右になりますが，下を向いていれば“右”は左になります。

このような処理をIF文で行っていると，4とおりのキャラクターの方向にさらに4とおりの方向の条件判断を行う必要があるため，条件判断処理が非常に多くなってしまいます。つまり，プログラムが長く，読みづらくなってしまうわけです。

しかし，方向を数値化していれば，この問題も解決します。**第2図**のように上下左右の方向をきめた場合，方向数値に1を加えれば“右”の方向が，3を加えれば“左”の方向が，それぞれ計算できるのです。なお，数値を加えることで方向数値が3を超えた場合は，4を引いてやります。キャラクターの向きが2（下）で，“左”を求める場合は3を加え，結果の5から4を引いた1（右）が，「下を向いているキャラクターの左方向」になるわけです。

これは，**第2図**の数値の対応が時計まわりになっていることから，1増えると方向が90度変わることを利用しています。ただし，左から上への部分だけは不連続になっているため，4を引くという補正をかけます（**第6図**）。あるいは，単純に4で割った“余り”を用いてもかまいません。

ベーマガ・ホームページ ダウンロードの方法

まずは，以下のアドレスにインターネット・アクセスします。

●http://www.basicmagazine.net/

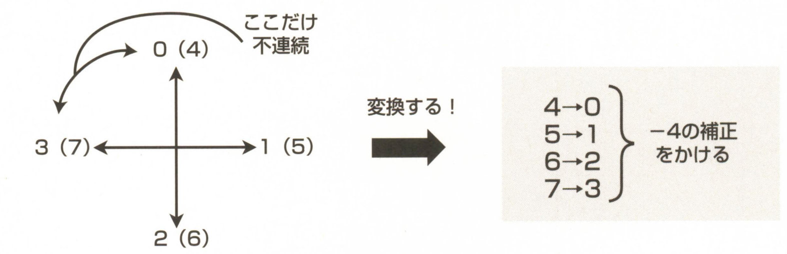
そこから，「連載プログラミング講座」をクリックしてください。

そして，『図解！ VBゲーム作成術』の画面の下にある「ファイルをダウンロード」をクリックします。すると，「ファイルのダウンロード」ウィンドウが表示されるので，「このファイルをディスクに保存する」をチェックして[OK]ボタンを押します。「名前を付けて保存」ウィンドウが表示されるので，適当な場所（c:¥bm11など）に保存します。

同じ場所に“lha.exe”をコピーして，[スタート]-[プログラム]-[MS-DOSプロンプト]で，その場所に移動します（上記の例だと「cd c:¥bm11」と入力）。そして，「lha e z2001-11」とキーボードから入力して，[Enter]キーを押します。これで，解凍作業は終わりました。

“readme.txt”には，ファイルの概要が書かれていますので，参照してください。

方向数値の補正



《第6図》上下左右に対応させた数値を，このように補正してやる

ゲームで処理を体感してみよう！

それでは，実際にVBでプログラムされたキャラクター・アクション・ゲームを実行してみましょう。プログラムは，**リスト1**です。

リストを入力して実行すると，**第7図**のような画面が表示されます。画面にはプレイヤーが操作するプレイヤー・キャラクターと，敵キャラクターが四つ登場します。敵キャラクターは，ランダムに格子状の通路を移動します。移動した後に，ドットを落としていく場合があります。

プレイヤーは，プレイヤー・キャラクターをテンキーの[2][4][6][8]で上下左右に移動させ，敵キャラクターが落としたドットを制限時間内になるべくたくさん拾い集めてください。ドットを取ると，その段階での残り時間に応じた得点が加算されます(残り時間が多いほど高得点になる)。

ただし，敵キャラクターと接触すると残り時間が大幅に減ってしまいます。

リストの中身を見てみよう

リスト1では，移動方向を数値化して取り扱うことによって，プログラム自体が簡略化されています。さらに，プレイヤー・キャラクターの移動がスムーズにできるようになっています。

敵キャラクターの移動方向については，eh配列変数に**第2図**の対応をした数値を格納しています。また，移動方向から各座標方向の移動量を求めるテーブルとして，配列変数dx（X座標方向）とdy（Y座標方向）を用意してあります。

単純にehの方向に敵キャラクターを移動させるだけならば，敵キャラクターの座標ex，eyについて，

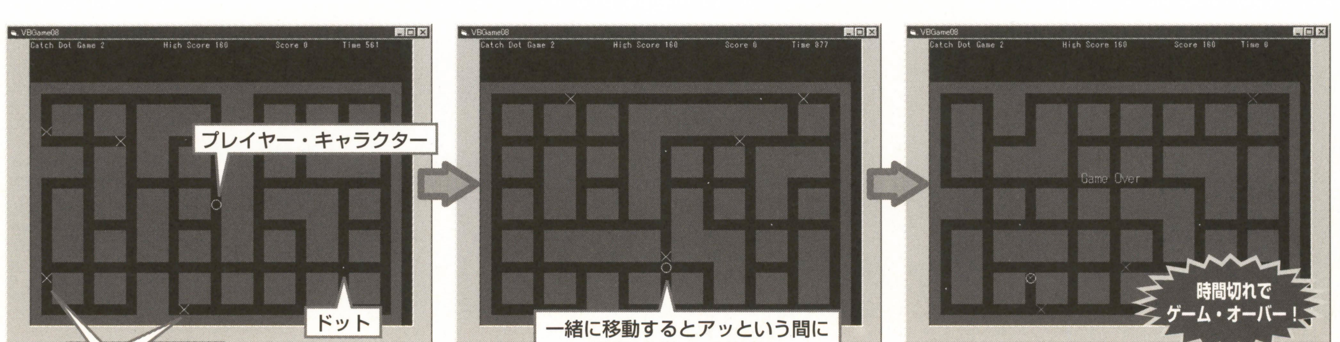
```
ex = ex + dx(eh)
ey = ey + dy(eh)
```

という処理を行うだけで済んでしまうのです。

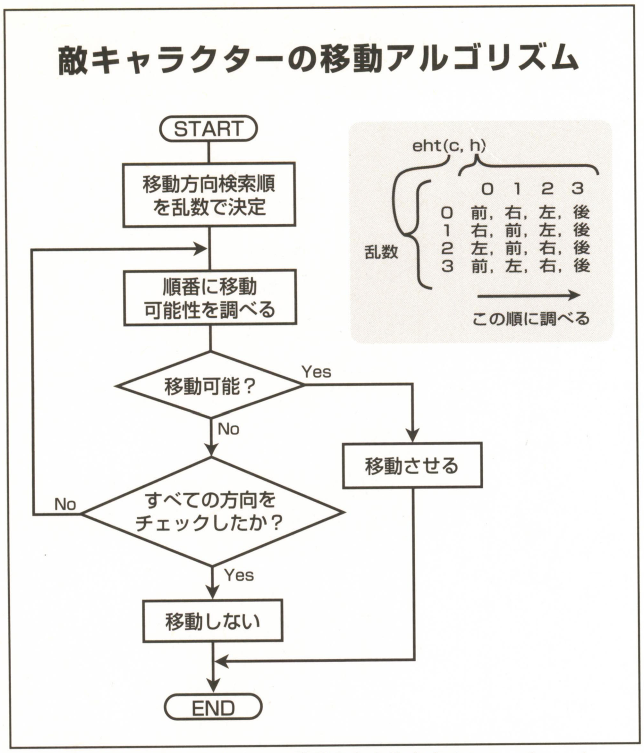
また，「方向→実際の移動量」の変換が簡単になっていることから，キャラクターの移動方向を決定する部分も簡単なプログラムになっています。敵キャラクター移動処理では，乱数によって移動方向検索テーブルehtを選択し，そのテーブルに記述された順番で移動可能かどうかを調べ，移動可能ならばその方向を選択するようになっています。

ゲーム・フィールドには壁があり，壁の方向には移動できません。このため，候補となる移動方向に移動できるかどうかを，仮想画面を使って確認する必要があります。ここで，移動方向の候補を変数hに用意してあるとすれば，移動候補

これが『キャッチ・ドットゲーム2』だ！



《第7図》プレイヤー・キャラクターの移動は，二つの移動キー入力を受けつけるようになっているため，交差点ではあらかじめ曲がりたい方向のキーも入力しておくとスムーズに曲がることができる。敵キャラクターと移動速度は同じなので，スムーズに曲がり続ける限り追いつかれることはない



《第8図》敵キャラクターの移動方向を決定するフローチャート

《第1表》『キャッチ・ドットゲーム2』の主な変数

変数名	内容
hiscore	ハイスコア
score	プレイヤー・スコア
px	プレイヤー・キャラクターX座標
py	プレイヤー・キャラクターY座標
ef ()	敵キャラクター状態フラグ
ex ()	敵キャラクターX座標
ey ()	敵キャラクターY座標
eh ()	敵キャラクターの方向
kg ()	仮想画面用2次元配列変数（0：空白，1：壁，2：ドット）
dx ()	方向→X座標移動量変換テーブル
dy ()	方向→Y座標移動量変換テーブル
eht ()	敵キャラクター移動方向検索順テーブル
ti	残り時間
dotc	ドットを発生させる間隔計測カウンタ
gamestat	ゲーム進行状況（0：ゲーム中，1：ゲーム・オーバー）

先の状況は仮想画面kgを，

```
kg(ex+dx(h), ey+dy(h))
```

として参照するだけでわかります。

敵キャラクターは，第8図のような移動アルゴリズムを使っています。これにより，移動可能な移動先の状況判断が簡単に処理できるうえ，“後ろ”という概念（移動方向数値に2を加えて4で割った余り）も簡単に取り扱えるために，プログラム的に単純な流れになっています。

一方，プレイヤー・キャラクターの移動時には，押されているキーによる移動希望方向と，それまでキャラクターが移動した方向の両方を考慮して，なめらかに移動するようにしてあります。これは，「二つの移動キーが押されていた場合，それまで移動してきた方向と，それと別の方向の両方について移動できるかどうかを判断し，新しく移動できる方向があればそちらを優先する」という処理で実現しています。

このような処理を行う場合にも，移動先の状況判断が複数回必要となりますが，それが簡単に処理できるためにリストはそれほど長くなっていません。

このように，移動方向を数値化すれば，方向をテーブルに組み込めるようになり，方向に関連する処理が非常に簡単になっていきます。この手法は，移動方向が4とおりのキャラクター・アクション・ゲームだけでなく，8とおりのゲームや，2とおりしかない平面シューティング・ゲームに適用することもできます。

さらに，方向を拡張することで，移動方向が多い立体タイプのゲームなどにも適用できます。

まとめ

コンピュータは，基本的に数値しか取り扱えません。しかし，数値にいろいろなものを対応させることで，数値以外の情報も処理できるようになっています。

キャラクター・アクション・ゲームの“方向”についても，直接処理できませんが，「方向→実際の移動量」の対応関係を用意することで，数値によって処理できるようになるわけです。

次回は，敵キャラクターにプレイヤー・キャラクターを追跡させるアルゴリズムを紹介します。お楽しみに！

《リスト1》『キャッチ・ドットゲーム2』のプログラム・リスト（VB6.0用）

```
Option Explicit

Dim keystat%(9) 'テンキーの状態を記録する変数
Dim keytrig0%, keytrig1%, keytrig2%, keytrig3% 'トリガー用キーの
状態を記録する変数
Dim px%, py%, ph%, pf% 'プレイヤーキャラクタ
Dim ex%(3), ey%(3), eh%(3), ef%(3) '敵キャラクタ
Dim kg(35, 26) '仮想画面
Dim eht(3, 3) '敵キャラクタ移動方向検索テーブル
Dim dx(3), dy(3) '方向→移動量変換テーブル
Dim hiscore%, score% 'ハイスコアとスコア
Dim ti% '残り時間
Dim dotc% 'ドットを発生させる間隔を数えるカウンタ
Dim gamestat% 'ゲーム進行状況

Private Sub Form_Load()
    hiscore = 0
    gameinit
End Sub

Private Sub gameinit() 'ゲーム初期化
    Dim ox%, oy%, x%, y%, xx%, yy%, i%, j%, f%, h%, c0%, c1%
    With Form1
        .Width = Screen.TwipsPerPixelX * 640 'フォームのサイズを
設定
        .Height = Screen.TwipsPerPixelY * 480
        .Caption = "VBGame08"
    End With
    With Picture1
        .Width = Screen.TwipsPerPixelX * 576 'ピクチャボックスの
サイズを設定
        .Left = Screen.TwipsPerPixelY * 32
        .Height = Screen.TwipsPerPixelY * 432
        .Top = 0
        .ScaleMode = 3
        .ScaleHeight = 432
        .ScaleWidth = 576
        .BackColor = RGB(0, 0, 0)
        .ForeColor = RGB(255, 255, 255)
        .FillStyle = 0
        .Font = "ＭＳ ゴシック"
        .FontSize = 10
    End With
    Picture1.Cls
    Form1.Show

    dx(0) = 0: dy(0) = -1 '移動方向→移動量のテーブル
    dx(1) = 1: dy(1) = 0
    dx(2) = 0: dy(2) = 1
    dx(3) = -1: dy(3) = 0

    eht(0, 0) = 0: eht(0, 1) = 1: eht(0, 2) = 3: eht(0, 3) = 2 '
敵キャラクタ移動方向検索順テーブル
    eht(1, 0) = 1: eht(1, 1) = 0: eht(1, 2) = 3: eht(1, 3) = 2
    eht(2, 0) = 3: eht(2, 1) = 0: eht(2, 2) = 1: eht(2, 3) = 2
    eht(3, 0) = 0: eht(3, 1) = 3: eht(3, 2) = 1: eht(3, 3) = 2

    Picture1.Print "Catch Dot Game 2"

    px = 17 'プレイヤーキャラクタの初期座標
    py = 15

    dotc = 100
    For i = 0 To 3
        ef(i) = 1 '存在フラグをたてる
        eh(i) = 0
    Next

    ex(0) = 1 '敵キャラクタの初期座標
    ey(0) = 5
    ex(1) = 33
    ey(1) = 5
    ex(2) = 1
    ey(2) = 25
    ex(3) = 33
    ey(3) = 25

    '最初に壁でゲームフィールドを埋める
    For y = 4 To 26
        For x = 0 To 34
            kg(x, y) = 1 '壁に対応するコード
            Picture1.Line (x * 16, y * 16)-(x * 16 + 15, y * 16 +
15), RGB(128, 0, 128), BF '壁表示
        Next
    Next

    '通路を作る
    For y = 5 To 25 Step 4
        For x = 1 To 33
            kg(x, y) = 0 '空白に対応するコード
            Picture1.Line (x * 16, y * 16)-(x * 16 + 15, y * 16 +
15), RGB(0, 0, 0), BF '空白（通路）表示
        Next
    Next
```

つぎのページに続く

→ リスト1の続き

```
    For x = 1 To 33 Step 4
        For y = 5 To 25
            kg(x, y) = 0 '空白に対応するコード
            Picture1.Line (x * 16, y * 16)-(x * 16 + 15, y * 16 +
15), RGB(0, 0, 0), BF '空白（通路）表示
        Next
    Next

    '一部の通路を壁に戻す
    For i = 0 To 14
        f = 0
        While f = 0
            f = 1
            x = Int(Rnd(1) * 9) * 4 + 1
            y = Int(Rnd(1) * 6) * 4 + 5
            h = Int(Rnd(1) * 4)
            c0 = 0
            c1 = 0
            For j = 0 To 3
                If kg(x + dx(j), y + dy(j)) = 0 Then
                    c0 = c0 + 1
                End If
                If kg(x + dx(h), y + dy(h)) = 0 Then
                     If kg(x + dx(h) * 4 + dx(j), y + dy(h) * 4 +
dy(j)) = 0 Then
                        c1 = c1 + 1
                    End If
                Else
                    c1 = 0
                End If
            Next
            If c0 < 3 Or c1 < 3 Then
                f = 0
            ElseIf kg(x + dx(h), y + dy(h)) <> 0 Then
                f = 0
            End If
        Wend
        For j = 1 To 3
            kg(x + dx(h) * j, y + dy(h) * j) = 1
            xx = x + dx(h) * j
            yy = y + dy(h) * j
             Picture1.Line (xx * 16, yy * 16)-(xx * 16 + 15, yy *
16 + 15), RGB(128, 0, 128), BF '壁表示
        Next
    Next

    'プレイヤーキャラクタの出現場所の周囲は強制的に通路にする
    x = 17
    For y = 13 To 17
        kg(x, y) = 0
          Picture1.Line (x * 16, y * 16)-(x * 16 + 15, y * 16 +
15), RGB(0, 0, 0), BF '空白（通路）表示
    Next

    score = 0
    ti = 1000
    putscore

     gamestat = 0 'ゲームモード（0でゲーム中、1でゲームオーバー
状態）

    Timer1.Interval = 70 'インターバルタイマ起動
End Sub

Private Sub Picture1_MouseUp(button%, shift%, x!, y!)
    If (button And 2) = 2 Then 'マウス右ボタンクリックで強制終了
        End
    End If
End Sub

Private Sub Picture1_KeyDown(key%, shift%)
    If key = 27 Then 'ESCキーで強制終了
        End
    End If
    If key > 96 And key < 106 Then 'テンキー設定
        keystat(key - 96) = 1
    End If
    If key = 32 Then keytrig0 = 1 'トリガキー設定
    If key = 90 Then keytrig1 = 1
    If key = 88 Then keytrig2 = 1
    If key = 13 Then keytrig3 = 1
End Sub

Private Sub Picture1_KeyUp(key%, shift%)
    If key > 96 And key < 106 Then
       keystat(key - 96) = 0
    End If
    If key = 32 Then keytrig0 = 0
    If key = 90 Then keytrig1 = 0
    If key = 88 Then keytrig2 = 0
    If key = 13 Then keytrig3 = 0
End Sub

Private Sub Timer1_Timer()
    Dim npx%, npy%, npvx%, npvy%, nph1%, nph2%, mf%, nex%, ney%
    Dim i%, f!, d%, h%, c%, k%, nh%
    If gamestat = 0 Then '通常ゲームモード
        'プレイヤーキャラクタの処理
        npx = px
        npy = py
        npvx = 0
        npvy = 0
        If keystat(2) = 1 Then '下キーを押しているか？
            npvy = npvy + 1
        End If
        If keystat(4) = 1 Then '左キーを押しているか？
            npvx = npvx - 1
        End If
        If keystat(6) = 1 Then '右キーを押しているか？
            npvx = npvx + 1
        End If
        If keystat(8) = 1 Then '上キーを押しているか？
            npvy = npvy - 1
        End If
        If npvx <> 0 Or npvy <> 0 Then '移動キーを押しているか？
            If npvx <> 0 And npvy <> 0 Then '移動キーを２つ押して
いるか？
                If ph = 0 Or ph = 2 Then
                    nph1 = 2 - npvx '第１候補
                    nph2 = 1 + npvy '第２候補
                Else
                    nph1 = 1 + npvy
                    nph2 = 2 - npvx
                End If
            Else
                If npvx <> 0 Then
                    nph1 = 2 - npvx '押されている移動キーが１つの
場合は第１候補と第２候補を同じものにする
                    nph2 = 2 - npvx
                Else
                    nph1 = 1 + npvy
                    nph2 = 1 + npvy
                End If
            End If
            If kg(npx + dx(nph1), npy + dy(nph1)) <> 1 Then '第１
候補の方向が壁で無いか？
                mf = 1
                ph = nph1
            ElseIf kg(npx + dx(nph2), npy + dy(nph2)) <> 1 Then '
第２候補の方向が壁で無いか？
                mf = 1
                ph = nph2
            Else
                mf = 0 '第１候補＆第２候補とも壁で移動できない場
合
            End If
            If mf = 1 Then '移動できる場合
                npx = npx + dx(ph) '移動できる方向に移動
                npy = npy + dy(ph)
                putplayer px, py, 0
                putplayer npx, npy, RGB(255, 255, 255)
                px = npx '座標更新
                py = npy
                If kg(px, py) = 2 Then '移動先にドットがあるか？
                    kg(px, py) = 0 '仮想画面のドットを消す
                    putdot px, py, 0
                    scoreup 10 + Int(ti / 100)    '得点アップ
                End If
            End If
        End If
        putplayer px, py, RGB(255, 255, 255)
        '敵キャラクタの処理
        For i = 0 To 3
            If ef(i) = 0 Then '存在していない場合
            ElseIf ef(i) = 1 Then '存在している場合
                If px = ex(i) And py = ey(i) Then 'プレイヤーキャ
ラクタとの衝突判定
                    pf = 1 'プレイヤーダメージフラグ
                End If
                c = Int(Rnd(1) * 4)  '移動方向検索順を乱数で決定
                d = -1
                 For h = 0 To 3 '移動方向テーブルによる方向を順番
にチェック
                    nh = eh(i) + eht(c, h)
                     If nh > 3 Then nh = nh - 4 '相対的方向のため
の補正
                    k = kg(ex(i) + dx(nh), ey(i) + dy(nh))
                     If k = 0 Or k = 2 Then '移動先が空白かドット
ならば移動
                        d = nh
                        h = 3
                    End If
                Next
                If d > -1 Then
                    eh(i) = d
                    nex = ex(i) + dx(eh(i)) '移動先の座標
                    ney = ey(i) + dy(eh(i))
                    putenemy ex(i), ey(i), RGB(0, 0, 0)
                    putenemy nex, ney, RGB(255, 0, 0)
                     If kg(ex(i), ey(i)) = 2 Then '元の場所にドッ
トがあった場合
                        putdot ex(i), ey(i), RGB(255, 255, 0) 'ド
ットを描き直す
                    End If
                     If dotc < 1 Then '一定間隔でドットを発生させ
る
                        putdot ex(i), ey(i), RGB(255, 255, 0) 'ド
ットを描く
                         kg(ex(i), ey(i)) = 2 '仮想画面にドットを
設定
                        dotc = 100
                    End If
                    dotc = dotc - 1 'ドット発生カウンタ更新
                    ex(i) = nex
                    ey(i) = ney
                    If px = ex(i) And py = ey(i) Then 'プレイヤー
キャラクタとの衝突判定
                        pf = 1 'プレイヤーダメージフラグ
                    End If
                End If
            End If
        Next
        If pf > 0 Then 'プレイヤーがダメージを受けている？
            ti = ti - 200 '残り時間を減らす
            pf = 0
        End If
        ti = ti - 1
        If ti < 0 Then ti = 0
        puttime
        If ti < 1 Then gamestat = 1 '時間切れでゲームオーバー
    ElseIf gamestat = 1 Then
        With Picture1
            .CurrentX = 230
            .CurrentY = 200
            .FontSize = 15
            .ForeColor = RGB(Rnd * 255, Rnd * 255, Rnd * 255)
        End With
        Picture1.Print "Game Over"
        If keytrig3 = 1 Then
            Timer1.Interval = 0 'インターバルタイマ停止
            gameinit
        End If
    End If
End Sub

Private Sub putscore()
    Picture1.Line (200, 0)-(369, 15), 0, BF
    With Picture1
        .CurrentX = 200
        .CurrentY = 0
        .FontSize = 10
        .ForeColor = RGB(255, 255, 255)
    End With
    Picture1.Print "High Score " & hiscore
    Picture1.Line (370, 0)-(469, 15), 0, BF
    With Picture1
        .CurrentX = 370
        .CurrentY = 0
        .FontSize = 10
        .ForeColor = RGB(255, 255, 255)
    End With
    Picture1.Print "Score " & score
End Sub

Private Sub puttime()
    Picture1.Line (470, 0)-(575, 15), 0, BF
    With Picture1
        .CurrentX = 470
        .CurrentY = 0
        .FontSize = 10
        .ForeColor = RGB(255, 255, 255)
    End With
    Picture1.Print "Time " & ti
End Sub

Private Sub scoreup(up)
    score = score + up
    If hiscore < score Then hiscore = score
    putscore
End Sub

Private Sub putplayer(x, y, c)
    Picture1.Circle (x * 16 + 8, y * 16 + 8), 7, c
End Sub

Private Sub putdot(x, y, c)
    Picture1.Line (x * 16 + 7, y * 16 + 7)-(x * 16 + 8, y * 16 +
8), c, BF
End Sub

Private Sub putenemy(x, y, c)
    Picture1.Line (x * 16, y * 16)-(x * 16 + 15, y * 16 + 15), c
    Picture1.Line (x * 16 + 15, y * 16)-(x * 16, y * 16 + 15), c
End Sub
```

リスト1入力＆実行方法

①VB起動

まずは，VBを起動します。新しいプロジェクトとして，「標準EXE」を選択して[開く]ボタンをクリックしてください。通常は，最初に標準EXEが選択された状態になっているので，[開く]をクリックするだけです。すると，**写真**①のような画面が表示されます。

②フォームの設定

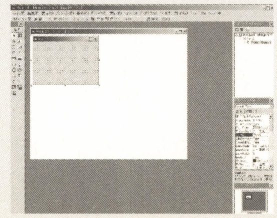
写真②のように，フォームを設定してください。プロパティは，プログラム中で更新されるので設定不要です（不用意に設定すると動作しなくなる場合があるので注意）。設定するのは，PictureBoxとTimerです。それぞれツールボックスのアイコンを選択した後に，フォーム上に貼り付けてください。貼り付ける操作は，フォーム上で左クリックした後，右下方向に少しドラッグしてから左ボタンを離せばOKです。位置，サイズなどは自由です（プログラム中で修正される）。

③リストの入力

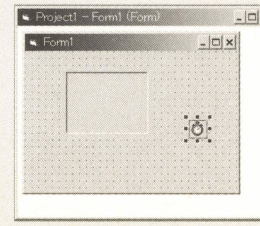
プロジェクト・ウィンドウのコード表示ボタンをクリックすると，コード・ウィンドウが表示されます。Form1のコードとして，**リスト1**を入力してください。

④実行

実行メニューから[開始]を選択すると，プログラムが実行されます。



《写真①》VBを起動したら「標準EXE」を選ぼう



《写真②》PictureBoxとTimerをフォームに配置する

プログラマー&クリエイターへの登竜門

BM特選
投稿プログラムコーナー

- Visual Basic
- Directx for Visual Basic
- HSP
- F-BASIC V6.0
- Q-Basic
- iアプリ
- PC-G850S
- MUSIC

本コーナーは，さまざまな言語に対応したプログラミング学習と創作をテーマに，読者からの投稿作品で成り立っています。これから，プログラミングを勉強しようというみなさんや，プロのプログラマー／クリエイターを目指そうというみなさんにとって役立つ情報が満載です。手近にあるパソコンに入力して，お楽しみください。

それぞれの記事にはプログラムを入力／実行するときの注意点やプログラムの内容，改造方法などが紹介されていますので，よく読んでください。また，“CHECKER FLAG”というコラムがあり，デバッグ博士ことDr.Dがそれぞれのプログラムに対し批評を加えています。口が悪い人なので，かなりキビシイことも言っていますが，自分でプログラムを作ったり，これから投稿しようとしている人にとっては，とても参考になるでしょう。



プログラム投稿者へのお願い

自分の作ったプログラムを投稿しようという人はつぎのきまりを守って応募してください。

①プログラム（作品）ファイル ②プログラムの説明，住所・氏名・年齢・職業（学校名）・電話番号（連絡のつく時間も）を書いた原稿のテキスト・ファイルをフロッピーディスクなどにセーブし（プリンタで打ち出した原稿も同封のこと），お送りください。

プログラムの説明には，①内容 ②使いかた・遊びかた ③操作方法 ④プログラムのしくみ（変数表やフローチャートも）⑤プログラムの具体的入力方法，注意点 ⑥改造のアドバイス，作ったとき苦労した点 ⑦使用機種名 ⑧言語（バージョン，モードなども）などを書きます。

注1） プログラムは必ずチェックし，ミスを防ぐために同じものを2～3入れてもらえば安心です。

注2） プログラムの説明は，テキスト・ファイルをフロッピーディスクなどにセーブし，念のためプリンタで打ち出したものも同封することが基本ですが，手書き原稿でも受け付けます。その際，できるだけていねいに，マス目のある用紙を使用してください。

注3） ディスクのラベル，封筒の裏面にもプログラム名（ファイル名も），原稿ファイル名，住所，氏名，電話番号を必ず記入してください。

内容はオリジナルのもので，未発表のものに限ります。もし，参考にした記事や本がありましたら，書名，号数，記事名，筆者名，記事のタイトルなども明記しておいてください。

お送りいただいたディスクはお返しできませんので，あらかじめバックアップをとっておいてください。

掲載されたプログラムに対しては，規定の原稿料をお支払いし，掲載誌をお送りいたします。

編集部

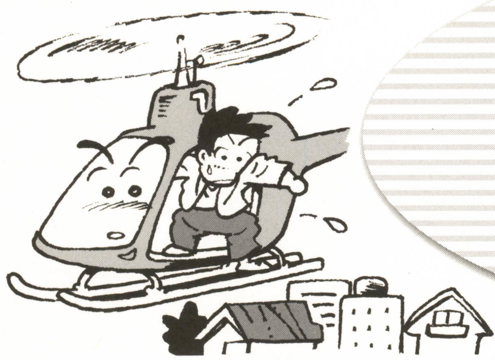
★本誌とじ込みの愛読者カードについている応募カードも忘れずに同封してください。

あて先：〒141-8715東京都品川区東五反田1-11-15　電波新聞社マイコンBASICマガジン「特選プログラム・コーナー」係

★ここに掲載されているプログラムは，ベーマガ・ホームページからもダウンロードできます。また，ホームページからの投稿も受け付けています。

http://www.basicmagazine.net/

Windows(Visual Basic V6.0)用



風向きの読みがカギになる

落下（らっか）

松本　輝彦（12歳）

内容

人をヘリコプターから降下させて，島に乗せるゲームです。

遊びかた・操作法

スペース・バーを押すとスタートします。カーソル・キーの左右でヘリを動かし，スペース・バーで人を降下させます。

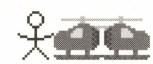
人は風に流されるので，風を考えて降下させてください。

ステージは全部で20ステージあります。全面クリアは結構難しいと思うので，がんばってください。

プログラムの仕組み

人と島の当たり判定は色で求めています。本当は計算で求めたかったのですが，わからなくなったので，この方法にしました。

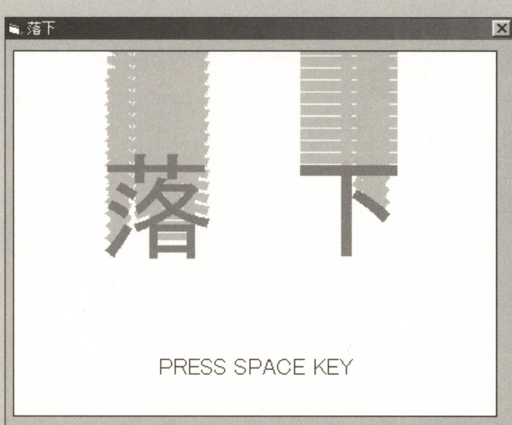
chara.gifの画像の大きさは，55×20です。人の大きさが15×20で，ヘリはそれぞれ20×20の大きさです。



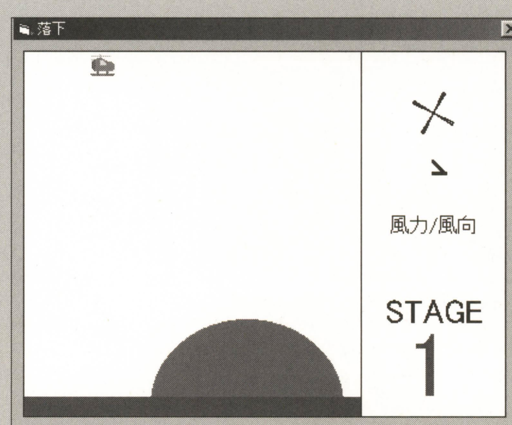
chara.gifの画像

苦労したところ

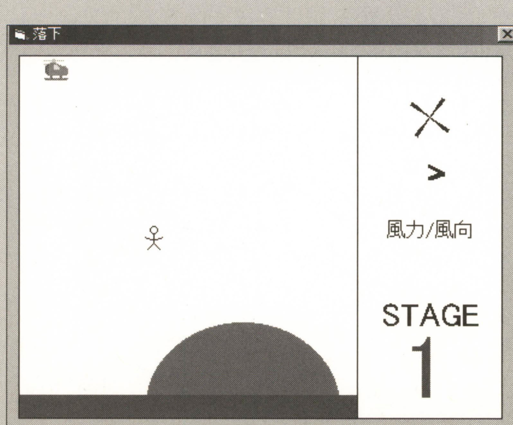
風力などの表しかたを考えるのに時間



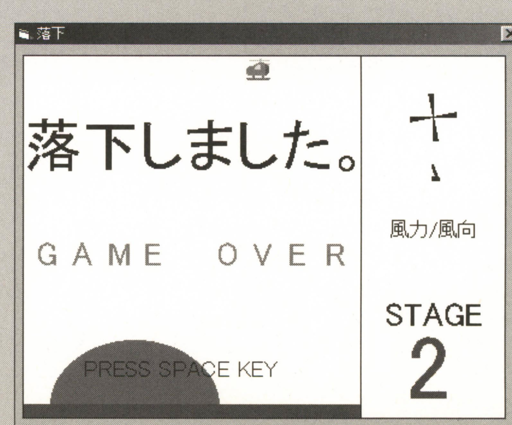
《写真1》人を落下させるゲームだ



《写真2》風を読んで適切な場所へ、ヘリを移動



《写真3》島へ着くように人を落下させる



《写真4》人が海へ行ってしまったら、ゲーム・オーバー

CHECKER FLAG

つぐ美：ゲーム内容，画像ともに単純ですが，けっこう遊べる作品です。風向きのシミュレーションが，しっかりできている印象だわ。

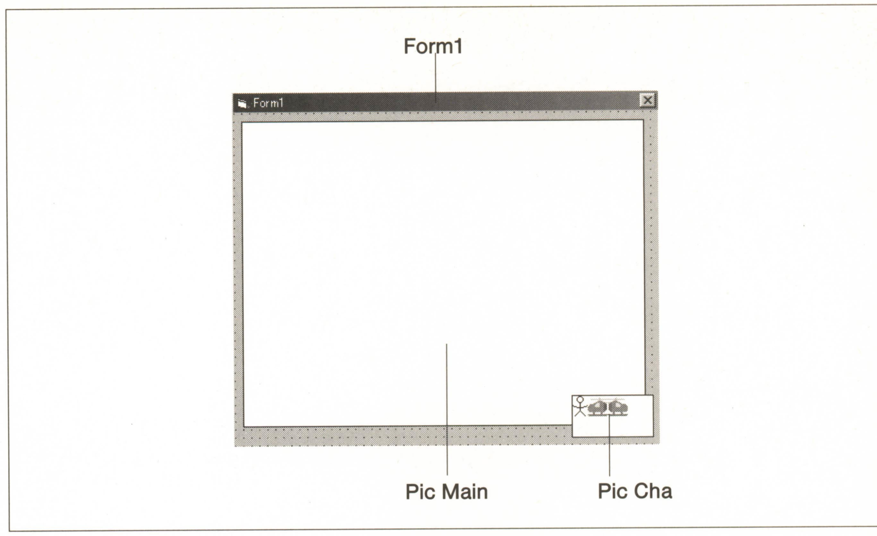
Dr.D：その風向きは，乱数を主に決定しているが，いくつかの式で計算を行っているため，動きのある風向きを再現できているというわけじゃ。この作品のリストは，その部分のコメントもしっかり記述してあるため，ぜひとも目を通してみてほしい。

編：リスト面でもう少し補足をすると，各ステージのデータをSelect～Case文で場合分けして代入するのは，チョット効率が悪いかな，と思います。配列に各ステージごとのデータをもっておいて，STを添字にして配列の値を代入するという手段のほうが，よりお手軽かもしれません。

がかかりました。下の矢印は上下方向の風も表しています。

最後に

同じような年齢の人が増えているので，負けないように頑張りたいと思っています。ホームページはhttp://kumamoto.cool.ne.jp/teru0820，メールアドレスは，mteruhiko@hotmail.comです。



フォームの配置図

《第1表》プロパティ表

フォームの設定	
Name	Form1
BorderStyle	1 '固定(実線)
Caption	Form1
Height	5310
Left	0
LinkTopic	Form1
MaxButton	0 'False
MinButton	0 'False
ScaleHeight	329
ScaleMode	3 'ピクセル
ScaleWidth	424
StartUpPosition	3 'Windows の既定値
Top	0
Width	6450
ピクチャボックスの設定	
Name	PicMain
Appearance	0 'フラット
AutoRedraw	-1 'True
BackColor	&H80000005&
ForeColor	&H80000008&
Height	302
Left	8
ScaleHeight	300
ScaleMode	3 'ピクセル
ScaleWidth	400
TabIndex	0
Top	8
Width	402
Name	PicCha
Appearance	0 'フラット
AutoRedraw	-1 'True
BackColor	&H80000005&
ForeColor	&H80000008&
Height	42
Left	336
Picture	chara.gif
ScaleHeight	40
ScaleMode	3 'ピクセル
ScaleWidth	80
TabIndex	1
Top	280
Visible	0 'False
Width	82

『落下』のプログラム・リスト

```
Option Explicit
Private Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, ByVal x As Long, ByVal y As
Long, ByVal nWidth As Long, ByVal nHeight As Long, ByVal hSrcDC As Long, ByVal xSrc As Long,
ByVal ySrc As Long, ByVal dwRop As Long) As Long
Private Declare Function GetAsyncKeyState Lib "user32" (ByVal vKey As Long) As Integer
Private Declare Function GetTickCount Lib "kernel32" () As Long
Const Rd = 3.14 / 180
'ステージ数
Const CS = 20
'ループ用、ステージ
Dim F As Integer, ST As Byte
'風車の角度、風の強さ
Dim FRK As Single, FP As Integer
'ヘリのX座標、ヘリの向き
Dim HX As Integer, HM As Byte
'人が落下しているか
Dim RH As Byte
'人のX座標、人のY座標
Dim ZX As Single, ZY As Single
'風力が変わる確率、落下するスピード
Dim KM As Byte, KU As Single
'島のX座標、島の幅、島の高さ
Dim AX As Integer, AW As Byte, AH As Byte
'波の高さ、波の高さが変わる確率、波の最大高さ
Dim SH As Integer, SK As Byte, SM As Byte
'最大風力
Dim FM As Byte, FN As String

'起動時の処理
```

```
Private Sub Form_Load()
Randomize
FN = App.Path & "¥"
'初期設定
ZX = 100: ZY = 0
RH = 0
HM = 1
ST = 1
Me.Show
Me.Caption = "落下"
'タイトル画面を描く
For F = 0 To 13
Call MSG("落　下", 70, -60 + F * 10, 7, 72)
Next F
Call MSG("落　下", 70, 80, 12, 72)
Call MSG("PRESS SPACE KEY", 120, 250, 0, 14)
'スペースバーが押されるまで待つ
While GetAsyncKeyState(32) <> 0: DoEvents: Wend
While GetAsyncKeyState(32) = 0: DoEvents: Wend
'メインルーチンへ
Call Main
End Sub
'メインルーチン
Sub Main()
'スペースバーが押されたか、クリアしたか
Dim SPB As Byte, CR As Byte
SPB = 0
'ステージを作る
Call Stage
'島のX座標、風力を決める
AX = Int(Rnd * (280 - AW * 2)) + AW
FP = Int(Rnd * 61) - 30
'現在のステージを一文字ずつ描く
PicMain.Cls
'「STAGE」
For F = 0 To 6
Call MSG(Right(Left("STAGE  ", F + 1), 1), F * 30 + 100, 50, 0, 28)
Call Wait(300)
Next F
'ステージ数
Call MSG(Str(ST), 250, 50, 0, 28)
Call Wait(1000)
'ここからメインループ
Do
DoEvents
'20ミリ秒待つ
Call Wait(20)
PicMain.Cls
'風車の角度を決める
FRK = (FRK + FP + 360) Mod 360
'カーソルキーが押されたらヘリを移動
'「←」
If GetAsyncKeyState(37) <> 0 And HX <> 0 Then HX = HX - 5: HM = 0
'「→」
If GetAsyncKeyState(39) <> 0 And HX <> 260 Then HX = HX + 5: HM = 1
'スペースバーが押されたら落下
If GetAsyncKeyState(32) <> 0 And SPB = 1 Then RH = 1

If RH = 0 Then ZX = HX + 2

If RH = 1 Then
   '人を落下させる
   ZY = ZY + KU
   'X座標の移動
   If ZY > 10 Then ZX = ZX + FP / 7
   '画面外に消えれば反対側にワープ
   If ZX < -20 Then ZX = 280
   If ZX > 280 Then ZX = -20
End If
'ある確率で風力変化
If Int(Rnd * (KM + 1)) = 0 Then FP = FP + Int(Rnd * 3) - 1
'ある確率で落下スピード変化
If Int(Rnd * (KM + 10)) = 0 Then KU = KU + (Int(Rnd * 3) - 1) / 5
'ある確率で波の高さ変化
If Int(Rnd * (SK + 1)) = 0 Then SH = SH + (Int(Rnd * 3) - 1)
'波の高さを調整
If SH < 0 Then SH = 0
If SH > SM Then SH = SM
'風力を調整
```


★C言語講座をもう一度開いてほしいものです。(福島県伊達郡・菅野友紀18歳) ……**【影:**Dr., C言語講座いつからスタートしましょうか。**Dr.D:**うむ, 講座開設に向け, より良い内容にするために綿密に打ち合わせ中じゃ。もうしばらく待っておくれ。**】**

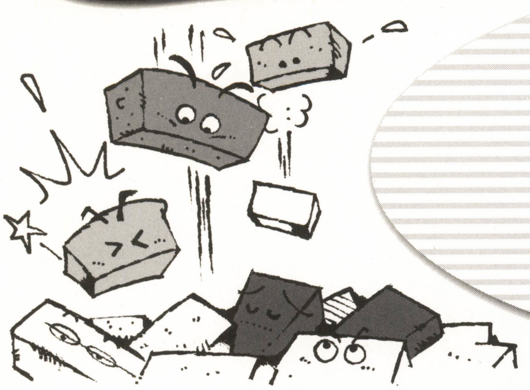
```
If FP < -FM Then FP = -FM
If FP > FM Then FP = FM
'落下スピード調整
If KU <= 0 Then KU = 0.1
'島を描く
PicMain.DrawWidth = 1
PicMain.FillStyle = 0
PicMain.FillColor = QBColor(2)
If AW < AH Then
PicMain.Circle (AX, 290), AH, QBColor(2), , , AH / AW
Else
PicMain.Circle (AX, 290), AW, QBColor(2), , , AH / AW
End If
'人を描く
If RH = 1 Then
  If ZX <= 265 Then
    BitBlt PicMain.hDC, ZX, ZY, 15, 20, PicCha.hDC, 0, 0, &H8800C6
  Else
    BitBlt PicMain.hDC, ZX, ZY, 280 - ZX, 20, PicCha.hDC, 0, 0, &H8800C6
  End If
End If
'ヘリを描く
BitBlt PicMain.hDC, HX, 0, 20, 20, PicCha.hDC, 15 + HM * 20, 0, &HCC0020
'海を描く
PicMain.FillColor = QBColor(9)
PicMain.Line (-5, 290 - SH)-(280, 300), QBColor(9), B
'右側の四角を書く
PicMain.FillColor = QBColor(15)
PicMain.Line (280, -5)-(400, 300), , B
'風車を描く
For F = -5 To 5
PicMain.Line (340, 50)-(340 + GetX(FRK - F, 20), 50 + GetY(FRK - F, 20))
PicMain.Line (340, 50)-(340 + GetX(FRK - F + 180, 20), 50 + GetY(FRK - F + 180, 20))
PicMain.Line (340, 50)-(340 + GetX(FRK - F + 90, 20), 50 + GetY(FRK - F + 90, 20))
PicMain.Line (340, 50)-(340 + GetX(FRK - F + 270, 20), 50 + GetY(FRK - F + 270, 20))
Next F
'右側の文字を描く
Call MSG("風力/風向", 305, 135, 0, 12)
Call MSG("STAGE", 300, 200, 0, 20)
Call MSG(Str(ST), 310 - (Len(Str(ST)) - 1) * 15, 220, 9, 56)
'矢印を描く
PicMain.ForeColor = QBColor(0)
PicMain.DrawWidth = 3
PicMain.Line (340, 100)-(340 + FP * 1.2, 100 - (2 - KU) * 10)
PicMain.DrawWidth = 2
PicMain.Line -(340, 90)
'島に乗っていればクリア
If PicMain.Point(ZX + 5, ZY + 10) = QBColor(2) Then CR = 1
SPB = 1
'人が波の下に隠れるか、クリアしていればループから抜ける
Loop Until ZY > 290 - SH Or CR = 1
'落下時
If ZY > 290 - SH Then
'文字を描く
Call MSG("落下しました。", 0, 50, 0, 36)
Beep
'750ミリ秒待つ
Call Wait(750)
'「GAME OVER」を一文字ずつ描く
For F = 0 To 8
Call MSG(Right(Left("GAME OVER", F + 1), 1), F * 30 + 10, 150, 12, 20)
Call Wait(150)
Next F
'1秒待つ
Call Wait(1000)
'文字を描く
Call MSG("PRESS SPACE KEY", 50, 250, 0, 14)
'ステージを1に戻す
ST = 1
'スペースバーが押されるまで待つ
While GetAsyncKeyState(32) <> 0: DoEvents: Wend
While GetAsyncKeyState(32) = 0: DoEvents: Wend
Else
'クリア時
'文字を描く
Call MSG("クリア!", 20, 50, 0, 56)
'次のステージに進める
ST = ST + 1
Beep
```

★家に5インチ・フロッピーとテープドライブが付いているパソコンがあります。確かROM-BASICだったと思ったけど（ちなみに20〜30年もの）。（新潟県三島郡・フリーゲーマー13歳）……【Dr.D：13歳にして化石みたいなマシンを知っているとは，恐れ入った。】

```
'2秒待つ
Call Wait(1000)
  'ゲームクリア時
  If ST - 1 = CS Then
  PicMain.Cls
  '「CONGRATULATION!」を一文字ずつ描く
  For F = 0 To 14
  Call MSG(Right(Left("CONGRATULATION!", F + 1), 1), F * 25 + 10, 100, 9, 28)
  Call Wait(150)
  Next F
  Call Wait(1000)
  Call MSG("PRESS SPACE KEY", 50, 250, 0, 14)
  'ステージを1に戻す
  ST = 1
  'スペースバーが押されるまで待つ
  While GetAsyncKeyState(32) <> 0: DoEvents: Wend
  While GetAsyncKeyState(32) = 0: DoEvents: Wend
  End If
End If
'人のY座標、落下判定をもどす
ZY = 0
RH = 0
Call Main
End Sub
'ステージ作成
Sub Stage()
Select Case ST
'島の幅、島の高さ、最大風力、風の変化率、落ちる速さ、波の最大高さ、波の変化率、波の高さ
Case 1: AW = 80: AH = 70: FM = 10: KM = 15: KU = 2: SM = 10: SK = 10: SH = 0
Case 2: AW = 70: AH = 55: FM = 8: KM = 10: KU = 2.5: SM = 8: SK = 15: SH = 0
Case 3: AW = 80: AH = 60: FM = 15: KM = 8: KU = 2: SM = 20: SK = 7: SH = 5
Case 4: AW = 80: AH = 55: FM = 10: KM = 10: KU = 1: SM = 15: SK = 7: SH = 5
Case 5: AW = 75: AH = 50: FM = 15: KM = 10: KU = 1: SM = 20: SK = 10: SH = 0
Case 6: AW = 50: AH = 80: FM = 20: KM = 5: KU = 2.3: SM = 20: SK = 7: SH = 10
Case 7: AW = 40: AH = 80: FM = 25: KM = 4: KU = 2.4: SM = 20: SK = 6: SH = 10
Case 8: AW = 40: AH = 75: FM = 30: KM = 3: KU = 2.5: SM = 25: SK = 5: SH = 15
Case 9: AW = 35: AH = 80: FM = 25: KM = 5: KU = 2: SM = 30: SK = 3: SH = 17
Case 10: AW = 35: AH = 70: FM = 30: KM = 1: KU = 2.4: SM = 35: SK = 1: SH = 20
Case 11: AW = 80: AH = 35: FM = 17: KM = 8: KU = 3: SM = 10: SK = 10: SH = 0
Case 12: AW = 80: AH = 35: FM = 20: KM = 6: KU = 2.7: SM = 15: SK = 7: SH = 5
Case 13: AW = 75: AH = 30: FM = 20: KM = 5: KU = 2.5: SM = 15: SK = 5: SH = 5
Case 14: AW = 75: AH = 25: FM = 20: KM = 7: KU = 1.8: SM = 15: SK = 7: SH = 7
Case 15: AW = 70: AH = 20: FM = 15: KM = 5: KU = 1: SM = 10: SK = 5: SH = 0
Case 16: AW = 60: AH = 40: FM = 10: KM = 10: KU = 3: SM = 30: SK = 1: SH = 0
Case 17: AW = 80: AH = 100: FM = 20: KM = 6: KU = 3: SM = 100: SK = 5: SH = 80
Case 18: AW = 50: AH = 50: FM = 15: KM = 1: KU = 0.5: SM = 20: SK = 10: SH = 5
Case 19: AW = 20: AH = 90: FM = 30: KM = 1: KU = 5: SM = 70: SK = 1: SH = 40
Case 20: AW = 40: AH = 30: FM = 30: KM = 1: KU = 1.8: SM = 20: SK = 1: SH = 10
End Select
End Sub
'文字を描く
Sub MSG(MS As String, MX As Integer, MY As Integer, FC As Byte, FS As Byte)
PicMain.CurrentX = MX
PicMain.CurrentY = MY
PicMain.ForeColor = QBColor(FC)
PicMain.FontSize = FS
PicMain.Print MS
End Sub
'Y座標の取得
Function GetY(ByVal Ag As Integer, ByVal Lg As Byte) As Single
GetY = Sin(Ag * Rd) * Lg
End Function
'X座標の取得
Function GetX(ByVal Ag As Integer, ByVal Lg As Byte) As Single
GetX = Cos(Ag * Rd) * Lg
End Function
'ウェイト
Sub Wait(T As Integer)
Dim At, ST
ST = GetTickCount
At = ST
While At - ST < T: DoEvents: At = GetTickCount: Wend
End Sub
'完全に終了
Private Sub Form_QueryUnload(Cancel As Integer, UnloadMode As Integer)
End
End Sub
```

★いまだに「BASIC」を「ばしっく」と読んでしまう不届き物…。(奈良県吉野郡・**田舎猫14歳**)……**【影：**本屋さんからの電話で，たまに「ばしっくマガジン」と注文してきますよね。**編：**「ベーマガ」を「ベースボールマガジン」とまちがえる人もいましたね。**】**

Windows(Visual Basic V6.0)用



落ち物パズルの基本形

スクウェア

Square

岡崎　務（19歳）

内容

上から落ちてくるブロックをうまく重ね合わせて，四つ以上同じ色のブロックが並ぶと消え，ひたすらハイスコアを狙うゲームです。

某ゲームによく似てますが気にしないでください（汗）。得点が一定以上になるとスピードレベルが上がります。



《写真１》VBで落ち物パズルを再現



《写真２》連鎖を狙っていくのが基本



《写真３》連鎖数が多いほど高得点



《写真４》うまくブロックを積んで，大連鎖を起こそう

遊びかた

フォームが現れたらリターン・キーでゲーム開始。

- **ブロックの移動：** カーソル・キー（上以外）。
- **ブロックの回転：** リターン・キーで右回り(?)，Lキーで左回り(?)。
- **その他：** スペース・キーで反転。

プログラムの入力

プロパティ表を見て，フォームにオブジェクトを貼り付けてプロパティを設定してください。あとはプログラムを入力するだけです。

注意するところ

ソースを見てもわかると思いますが，KeyDownの部分でそのまま処理してもできますが，あえてKeyDown部分ではボタンの情報だけを取っています。

ボタン情報をとらない場合は，ボタンを押したとき処理できるかどうか判定する必要があります。

改造など

背景やブロックをBitBltで転送して表示していくと奇麗になると思います。

苦労したところ

再帰処理で結構苦労しました。あとはそれほど難しくないと思います。

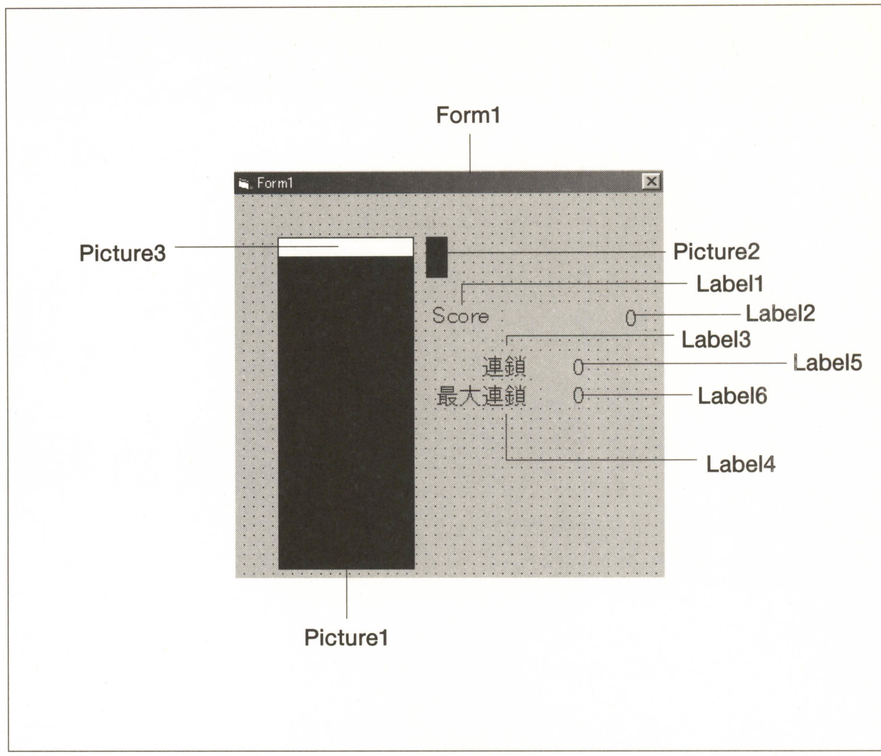
★編さん，バイトありがとう。そして影さんに左手を呪われた。痛いー。タスケテケレー。（愛知県一宮市・幻18歳）……【影：なに！編くんはアルバイト斡旋でもしてるのか！編：え？い，いや，身に覚えがないんですが。影：隠さないで，私にもお願いしますよ。】

CHECKER FLAG

影：見た目どおり，落ち物パズルの典型的な形のゲーム。独自性はハッキリ言ってゼロだけど，こういうスタンダードな作品は，ゲームを作りたいと思っているプログラム初心者には，参考になるんじゃないかな？

Dr.D：影クンの言うとおり，スタンダードな落ち物パズルのため，リスト面においても，キー入力部分以外は特別変わった記述はなく，シンプルな構成と言える。コメントがしっかり記述されているのは，ありがたいのぉ。

つぐ美：パソコンの環境によるのかもしれないけど，ブロックの色が暗めで見にくいわね。ゲームプログラムは，こういったプレイに関係する事柄のチョットした気配りも大事よ。



フォームの配置図

《第１表》プロパティ表

フォームの設定	
Name	Form1
BorderStyle	1 '固定(実線)
Caption	Form1
Height	5115
KeyPreview	-1 'True
Left	0
LinkTopic	Form1
LockControls	-1 'True
MaxButton	0 'False
MinButton	0 'False
ScaleHeight	316
ScaleMode	3 ピクセル
ScaleWidth	360
StartUpPosition	2 '画面の中央
Top	0
Width	5490
ピクチャボックスの設定	
Name	Picture1
Appearance	0 フラット
AutoRedraw	-1 'True
BackColor	&H00000000&
ForeColor	&H80000008&
Height	274
Left	37
ScaleHeight	272
ScaleMode	3 ピクセル
ScaleWidth	112
TabIndex	0
Top	35
Width	274
Name	Picture2
Appearance	0 フラット
AutoRedraw	-1 'True
BackColor	&H00000000&
ForeColor	&H80000008&
Height	34
Left	161
ScaleHeight	32
ScaleMode	3 ピクセル
ScaleWidth	16
TabIndex	1
Top	35
Width	18
Name	Picture3
Appearance	0 フラット
AutoRedraw	-1 'True
BackColor	&H80000005&
ForeColor	&H00000000&
Height	17
Left	37
ScaleHeight	15
ScaleMode	3 ピクセル
ScaleWidth	112
TabIndex	8
Top	35
Width	114
ラベルの設定	
Name	Label1
Caption	Score
Font	ＭＳ Ｐゴシック
…Size	14
Height	20
Left	167
TabIndex	2
Top	90
Width	54
Name	Label2
Alignment	1 '右揃え
Caption	0
Font	ＭＳ Ｐゴシック
…Size	14
Height	20
Left	232
TabIndex	3
Top	93
Width	106
Name	Label3
Caption	連鎖
Font	ＭＳ Ｐゴシック
…Size	14
Height	20
Left	207
TabIndex	4
Top	132
Width	44
Name	Label4
Caption	最大連鎖
Font	ＭＳ Ｐゴシック
…Size	14
Height	20
Left	169
TabIndex	5
Top	156
Width	86
Name	Label5
Alignment	1 '右揃え
Caption	0
Font	ＭＳ Ｐゴシック
…Size	14
Height	20
Left	256
TabIndex	6
Top	133
Width	38
Name	Label6
Alignment	1 '右揃え
Caption	0
Font	ＭＳ Ｐゴシック
…Size	14
Height	20
Left	257
TabIndex	7
Top	156
Width	37

★今日は締め切り当日…ふう。危ない危ない。…って，貴重なOF投稿欄が！（千葉県松戸市・**ほおく26歳**）……【**編：**影さん，どこ行っちゃったんでしょう。**編集長：**締切日になると身を隠すからな。今日はみんなで，飲みにでも行くか。**影：**賛成！**編集長：**見っけ！】

《リスト1》『Square』のプログラム・リスト

```
Private Declare Function GetTickCount Lib "kernel32" () As Long

Dim imap(16, 6) As Integer          '繋がっているかどうかの判定用
Dim map(16, 6) As Integer           'ブロック情報の配列
Dim otimap(16, 6) As Integer        '落ちているか落ちていないか判定用
Dim xy(1, 1) As Integer             '現在の位置
Dim co(3) As Integer                'ブロックの色情報
Dim cnt As Integer                  '繋がっている数
Dim T1 As Long                      '時間用
Dim T2 As Long                      '時間用
Dim OTITIME1 As Long                '下移動用
Dim OTITIME2 As Long                '下移動用
Dim sc As Long                      'スコア
Dim jt As Integer                   'ブロック状態(縦・横)
Dim mr As Integer, r As Integer     '最大連鎖数,連鎖数
Dim gf As Integer                   'ゲーム開始判定
Dim lv As Integer                   'レベル
Dim st As Integer, ost As Integer   '矢印情報と履歴(←  ↓  →  または押されてない)
Dim tr As Integer, otr As Integer   '回転情報と履歴(Return KeyとL Keyまたは押されてない)

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    If gf = 0 Then
     If KeyCode = vbKeyReturn Then Call init '開始
    Else
        Select Case KeyCode
            Case vbKeyDown:
                st = 2
            Case vbKeyLeft:
                st = 4
            Case vbKeyRight:
                st = 6
            Case vbKeySpace:
                st = 9: tr = 9
            Case vbKeyReturn:
                tr = 1
            Case vbKeyL
                tr = 2
        End Select
    End If
End Sub

Sub ido()
    OTITIME2 = GetTickCount()
    '矢印キーを押し続けても100ms以内は処理しない CPUによって各自変える
    If OTITIME2 - OTITIME1 <= 100 Then Exit Sub
    OTITIME1 = GetTickCount()

    Select Case st
        Case 2:         '下に移動
            If xy(0, 1) + 1 > 16 Then Exit Sub '範囲外のときは処理を抜ける
            If xy(1, 1) + 1 > 16 Then Exit Sub
            Select Case jt Mod 2 '下に物があると処理を抜ける
                Case 0: '縦の状態
                    If xy(0, 1) > xy(1, 1) Then
                        If map(xy(0, 1) + 1, xy(0, 0)) > 0 Then Exit Sub
                    Else
                        If map(xy(1, 1) + 1, xy(1, 0)) > 0 Then Exit Sub
                    End If
                Case 1: '横の状態
                    If map(xy(0, 1) + 1, xy(0, 0)) > 0 Then Exit Sub
                    If map(xy(1, 1) + 1, xy(1, 0)) > 0 Then Exit Sub
            End Select
            '下に移動
            xy(0, 1) = xy(0, 1) + 1
            xy(1, 1) = xy(1, 1) + 1
        Case 4:         '左に移動
            If xy(0, 0) - 1 < 0 Then Exit Sub '範囲外のときは処理を抜ける
            If xy(1, 0) - 1 < 0 Then Exit Sub
            Select Case jt Mod 2 '左に物があると処理を抜ける
                Case 0: '縦の状態
                    If map(xy(0, 1), xy(0, 0) - 1) > 0 Then Exit Sub
                    If map(xy(1, 1), xy(1, 0) - 1) > 0 Then Exit Sub
                Case 1: '横の状態
                    If xy(0, 0) < xy(0, 1) Then
                        If map(xy(0, 1), xy(0, 0) - 1) > 0 Then Exit Sub
                    Else
                        If map(xy(1, 1), xy(1, 0) - 1) > 0 Then Exit Sub
                    End If
            End Select
            '左に移動
            xy(0, 0) = xy(0, 0) - 1
            xy(1, 0) = xy(1, 0) - 1
        Case 6:         '右に移動
            If xy(0, 0) + 1 > 6 Then Exit Sub '範囲外のときは処理を抜ける
            If xy(1, 0) + 1 > 6 Then Exit Sub

            Select Case jt Mod 2 '右に物があると処理を抜ける
                Case 0: '縦の状態
                    If map(xy(0, 1), xy(0, 0) + 1) > 0 Then Exit Sub
                    If map(xy(1, 1), xy(1, 0) + 1) > 0 Then Exit Sub
                Case 1: '横の状態
                    If xy(0, 0) > xy(0, 1) Then
                        If map(xy(0, 1), xy(0, 0) + 1) > 0 Then Exit Sub
                    Else
                        If map(xy(1, 1), xy(1, 0) + 1) > 0 Then Exit Sub
                    End If
            End Select
            '右に移動
            xy(0, 0) = xy(0, 0) + 1
            xy(1, 0) = xy(1, 0) + 1
End Select
End Sub

Sub Change(T As Integer)
    Dim djt As Integer
    '右回りするときは状態に＋1
    If T = 1 Then
        If jt + 1 <> 4 Then
            djt = jt + 1
        Else
            djt = 0
        End If
    End If
    '左回りするときは状態－1
    If T = 2 Then
        If jt - 1 <> -1 Then
            djt = jt - 1
        Else
            djt = 3
        End If
    End If

    Select Case djt
        Case 0
            If xy(1, 1) + 1 > 16 Then
                    xy(1, 0) = xy(0, 0)
                    xy(1, 1) = xy(0, 1)
                    xy(0, 1) = xy(0, 1) - 1
            Else
                If map(xy(1, 1) + 1, xy(1, 0)) > 0 Then
                    xy(1, 0) = xy(0, 0)
                    xy(1, 1) = xy(0, 1)
                    xy(0, 1) = xy(0, 1) - 1
                Else
                    xy(1, 0) = xy(0, 0)
                    xy(1, 1) = xy(0, 1) + 1
                End If
            End If
        Case 1
            If xy(1, 0) + 1 > 6 Then
                If map(xy(0, 1), xy(0, 0) - 1) > 0 Then Exit Sub 'ブロックがあれば抜ける
                xy(1, 0) = xy(0, 0)
                xy(1, 1) = xy(0, 1)
                xy(0, 0) = xy(0, 0) - 1
            Else
                If map(xy(0, 1), xy(0, 0) + 1) > 0 Then
                 If (xy(0, 0) - 1 < 0) Then Exit Sub  '範囲外なら抜ける
                 If map(xy(0, 1), xy(0, 0) - 1) > 0 Then Exit Sub 'ブロックがあれば抜ける
```

★CMM試聴室に，「DirectMusic Segmentファイル」（牛食えやDDで使っているもの）で投稿してもいいのでしょうか。（新潟市・**佐藤良秋20歳**）……【**編：**もちろんOKです。どしどし投稿してくださいな。**つぐ美：**プログラムのほうも，何でも受け付けてますよ。】

```
                xy(1, 0) = xy(0, 0)
                xy(1, 1) = xy(0, 1)
                xy(0, 0) = xy(0, 0) - 1
            Else
                xy(1, 0) = xy(0, 0) + 1
                xy(1, 1) = xy(0, 1)
            End If
        End If
    Case 2
        If xy(1, 1) - 1 < 0 Then
            xy(1, 0) = xy(0, 0)
            xy(1, 1) = xy(0, 1)
            xy(0, 1) = xy(0, 1) + 1
        Else
            If map(xy(1, 1) - 1, xy(1, 0)) > 0 Then
                xy(1, 0) = xy(0, 0)
                xy(1, 1) = xy(0, 1)
                xy(0, 1) = xy(0, 1) + 1
            Else
                xy(1, 0) = xy(0, 0)
                xy(1, 1) = xy(0, 1) - 1
            End If
        End If
    Case 3
        If xy(1, 0) - 1 < 0 Then
            If map(xy(0, 1), xy(0, 0) + 1) > 0 Then Exit Sub 'ブロックがあれば抜ける
            xy(1, 0) = xy(0, 0)
            xy(1, 1) = xy(0, 1)
            xy(0, 0) = xy(0, 0) + 1
        Else
            If map(xy(0, 1), xy(0, 0) - 1) > 0 Then
              If (xy(0, 0) + 1 > 6) Then Exit Sub  '範囲外なら抜ける
              If map(xy(0, 1), xy(0, 0) + 1) > 0 Then Exit Sub 'ブロックがあれば抜ける
              xy(1, 0) = xy(0, 0)
              xy(1, 1) = xy(0, 1)
              xy(0, 0) = xy(0, 0) + 1
            Else
              xy(1, 0) = xy(0, 0) - 1
              xy(1, 1) = xy(0, 1)
            End If
        End If
    End Select
    '仮の状態から状態へ代入
    jt = djt
End Sub

Sub init()
    Dim i As Integer
    Dim j As Integer
    Dim c As Integer
    lv = 1: sc = 0 'レベルとスコアを初期化
    Randomize

    'ランダム色決定
    For i = 2 To 3
        c = Rnd(1) * 120
        Select Case c
            Case Is < 30
                co(i) = 1
            Case Is < 60
                co(i) = 2
            Case Is < 90
                co(i) = 3
            Case Is < 120
                co(i) = 4
        End Select
        '次のブロックを描画
        Picture2.Line (0, i * 16)-(15, i * 16 + 15), QBColor(co(i)), BF
    Next

     '配列の初期化
    For i = 0 To 16: For j = 0 To 6
        map(i, j) = 0: otimap(i, j) = 0
    Next: Next
    gf = 1 'ゲームフラグオン
    Call gmmain
End Sub

Private Sub Form_KeyUp(KeyCode As Integer, Shift As Integer)
    Select Case KeyCode
        Case vbKeyDown, vbKeyLeft, vbKeyRight   '↓  ←  →
            st = 0
        Case vbKeyReturn, vbKeyL                'Return か L
            tr = 0
        Case vbKeySpace                         'Space
            st = 0: tr = 0
    End Select
End Sub

Private Sub Form_Load()
    gf = 0: mr = 0 'ゲームフラグと最大連鎖数の初期化
End Sub

Private Sub Form_Unload(Cancel As Integer)
    End
End Sub

Sub gmmain()
    Dim i As Integer
    Dim j As Integer
    Do      'ブロックがいっぱいになったら抜ける
        'ブロック位置の初期化
        xy(0, 0) = 2: xy(0, 1) = 0
        xy(1, 0) = 2: xy(1, 1) = 1
        '色情報の移動
        co(0) = co(2): co(1) = co(3)

        '次の色決定
        For i = 0 To 1
            c = Rnd(1) * 120
            Select Case c
                Case Is < 30
                    co(i + 2) = 1
                Case Is < 60
                    co(i + 2) = 2
                Case Is < 90
                    co(i + 2) = 3
                Case Is < 120
                    co(i + 2) = 4
            End Select
            map(xy(i, 1), xy(i, 0)) = co(i)
            '次のブロックを描画
            Picture2.Line (0, i * 16)-(15, i * 16 + 15), QBColor(co(i + 2)), BF
        Next
        T1 = GetTickCount() 'ウェイト１を取得
        jt = 0: r = 0       '状態と連鎖数をリセット
        Do 'ブロックが下に落ちたら抜ける
        Call drow                           '四角を描画

        If st <> 0 Or tr <> 0 Then
            '現在位置の色情報を取得し配列に  0  を入れておく
            c1 = map(xy(0, 1), xy(0, 0))
            c2 = map(xy(1, 1), xy(1, 0))
            map(xy(0, 1), xy(0, 0)) = 0
            map(xy(1, 1), xy(1, 0)) = 0

            If st <> 9 And tr <> 9 Then
                If st <> 0 Then Call ido
                If tr <> 0 And otr = 0 Then Call Change(tr)
            Else
                'スペースキーで反転させる・・・
                If otr = 0 Then dm = c1: c1 = c2: c2 = dm
            End If
            '色情報を現在位置に戻す
            map(xy(0, 1), xy(0, 0)) = c1
            map(xy(1, 1), xy(1, 0)) = c2
        End If
        ost = st: otr = tr

        If xy(0, 1) <> 16 And xy(1, 1) <> 16 Then
```

★いつも汚い字ですみません。こんな字でも読んでいただいてると思うと，ヒデキ感激です。（茨城県土浦市・ヒデキ15歳）……**【編：ヘンも感激！字が汚くたって，丁寧に書いてくれているので，大丈夫ですよ。ちゃんと目を通しています。OF掲載は初めてかな？】**

```
            Select Case jt Mod 2
                Case 0: '縦の状態
                    If xy(0, 1) > xy(1, 1) Then
                        If map(xy(0, 1) + 1, xy(0, 0)) > 0 Then Exit Do
                    Else
                        If map(xy(1, 1) + 1, xy(1, 0)) > 0 Then Exit Do
                    End If
                Case 1: '横の状態
                    If map(xy(0, 1) + 1, xy(0, 0)) > 0 Then Exit Do
                    If map(xy(1, 1) + 1, xy(1, 0)) > 0 Then Exit Do
            End Select

            T2 = GetTickCount() 'ウェイト2を取得

            'CPUでウェイトの数（↓）を変える
            If T2 - T1 > 800 - lv * 10 Then
                '色情報を入れ換え現在位置をずらす
                c1 = map(xy(0, 1), xy(0, 0))
                c2 = map(xy(1, 1), xy(1, 0))
                map(xy(0, 1), xy(0, 0)) = 0
                map(xy(1, 1), xy(1, 0)) = 0
                map(xy(0, 1) + 1, xy(0, 0)) = c1
                map(xy(1, 1) + 1, xy(1, 0)) = c2
                xy(0, 1) = xy(0, 1) + 1
                xy(1, 1) = xy(1, 1) + 1
                T1 = GetTickCount()
            End If
        Else
            T2 = GetTickCount() 'ウェイト2を取得
            'CPUでウェイトの数（↓）を変える
            If T2 - T1 > 800 - lv * 10 Then Exit Do
        End If
        DoEvents
    Loop

    Call drow                          '四角を描画

    Do '消えなくなったら抜ける
        Do '落ちなくなったら抜ける
            c = 0 '処理をするかどうかのフラグをリセット
            For i = 16 To 1 Step -1: For j = 0 To 6
                '1つ下に何も無ければ落ちるフラグを立て
                'ブロック情報を1つ下に移動させる
                If map(i, j) = 0 And map(i - 1, j) > 0 Then
                    otimap(i - 1, j) = 1          '落ちるフラグを立てる
                    map(i, j) = map(i - 1, j)
                    map(i - 1, j) = 0
                    c = 1                      '処理をするかどうかのフラグ立てる
                End If
            Next: Next

            If c = 1 Then
                Do '全て移動し終えると抜ける
                    c = 0                      '処理をするかどうかのフラグをリセット
                    For i = 15 To 0 Step -1: For j = 0 To 6
                     If otimap(i, j) > 0 Then
                      '色なし
                      Picture1.Line (j * 16, i * 16 + (otimap(i, j) - 1))-(j * 16 + 15, i *
16 + (otimap(i, j) - 1) + 15), QBColor(0), BF
                      '色あり
                      Picture1.Line (j * 16, i * 16 + otimap(i, j))-(j * 16 + 15, i * 16 +
otimap(i, j) + 15), QBColor(map(i + 1, j)), BF
                      otimap(i, j) = otimap(i, j) + 1
                      If otimap(i, j) = 17 Then otimap(i, j) = 0 '落ちフラグをリセット
                      c = 1                 '処理をするかどうかのフラグを立てる
                     End If
                    Next: Next
                    wait (1)                   'ウェイト　CPUによって各自で変える
                    DoEvents
                Loop Until c = 0
                c = 1                          '処理をするかどうかのフラグを戻す
            End If
        Loop Until c = 0

        'チェックして消す処理
        For i = 1 To 16: For j = 0 To 6
            If map(i, j) <> 0 Then
                Call checkinit                     'チェック前の準備
                Call checkmain(j, i, map(i, j))    '再帰関数呼び出し
                If cnt >= 4 Then                   '繋がっている数が4つ以上のとき
                    Call elese                     '消す
                    c = 1                          '処理をするかどうかのフラグを立てる
                    r = r + 1                      '連鎖数をプラス
                    sc = sc + r * 100              'スコアをプラス
                End If
            End If
        Next: Next
        Call drow                          '四角を描画

        Label2.Caption = sc                'スコアを表示
        DoEvents
    Loop Until c = 0
    lv = Int(sc / 5000) + 1                'レベル設定
    If lv > 50 Then lv = 50                'レベルの最高値
    If r > mr Then mr = r                  '最大連鎖数と連鎖数の比較・入れ換え
    Label5.Caption = r                     '連鎖数
    Label6.Caption = mr                    '最大連鎖数
    DoEvents
  Loop Until map(1, 2) <> 0
  gf = 0 'ゲームフラグオフ
End Sub

Sub wait(T As Integer)
  Dim WT1, WT2
  WT1 = GetTickCount()
  Do
    WT2 = GetTickCount()
    DoEvents
  Loop Until WT2 - WT1 > T
End Sub

Sub checkmain(x As Integer, y As Integer, c As Integer)
  '配列の範囲を超えると関数から抜ける
  If x = -1 Or y = 0 Or x = 7 Or y = 17 Then Exit Sub
  '色が違っていたら抜ける
  If map(y, x) <> c Then Exit Sub
  '一度処理した場所は抜ける
  If imap(y, x) > 0 Then Exit Sub
  '処理した場所にフラグを立てる：繋がっている数を足す
  imap(y, x) = 1: cnt = cnt + 1

  Call checkmain(x + 1, y, c) '右
  Call checkmain(x - 1, y, c) '左
  Call checkmain(x, y - 1, c) '上
  Call checkmain(x, y + 1, c) '下
End Sub

Sub drow()
  '四角を描画するための処理
  Dim i, j As Integer
  For i = 1 To 16: For j = 0 To 6
    Picture1.Line (j * 16, i * 16)-(j * 16 + 15, i * 16 + 15), QBColor(map(i, j)), BF
  Next: Next
End Sub

Sub checkinit()
  'ブロックチェックの準備
  Dim i, j As Integer
  For i = 1 To 16: For j = 0 To 6
    imap(i, j) = 0
  Next: Next
  cnt = 0
End Sub

Sub elese()
  Dim i, j As Integer
  For i = 1 To 16: For j = 0 To 6
    If imap(i, j) > 0 Then map(i, j) = 0 '消えた部分をリセット
  Next: Next
End Sub
```

★いつも楽しみにしています。マイコンという名があるのですから，PICやZ86やH8などのプログラム記事もお願いします。とくに，WinCEやPalmのプログラムなどもやってください。（栃木県宇都宮市・**山本昇一41歳**）……**【編：前向きに考えているところです。】**

Windows(Visual Basic V6.0)用



基本タイピングの２本立て

普通(ふつう)のタイピング・ゲーム

高橋　好人（17歳）

内 容

タイトルは『普通のタイピング・ゲーム』です…。短さがウリのタイピング・ゲームです。ベーマガと言えば，超ソースが短いプログラムです。VBで極限まで削ったタイピング・ゲームです。短さは唖然とさせるでしょう…。

作品は二つあり，Project1は短さに絞ったゲーム，Project2はProject1を改造して，制限時間を付けたものです。

遊びかた

基本的に左側の文字を打ち続ける。

Project1はひたすら打ち続けてください。Project2ははじめに何かキーを押して60秒間ひたすら打ち続けてください。

入力方法

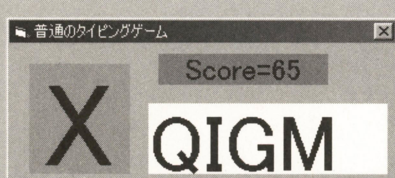
・Project1：フォームにPS，SC，SSという名のラベルを貼り付けてソースを入力します。

・Project2：フォームにPS，SC，SS，DS，TSという名のラベルと，Timer1という名のタイマーを貼り付け，インターバルを1000にしてソースを入力します。

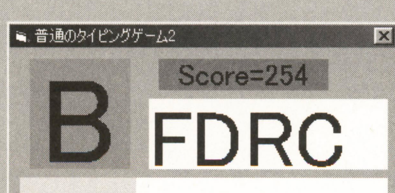
改 造

Project2：Const TMax=60 の値を変えるとプレイ時間変更が可能。

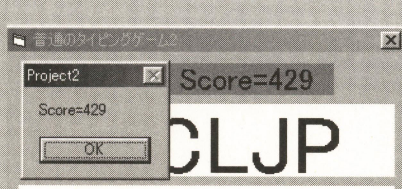
Project1・2：Score=Score+5の下にBeepと入れてみてください。



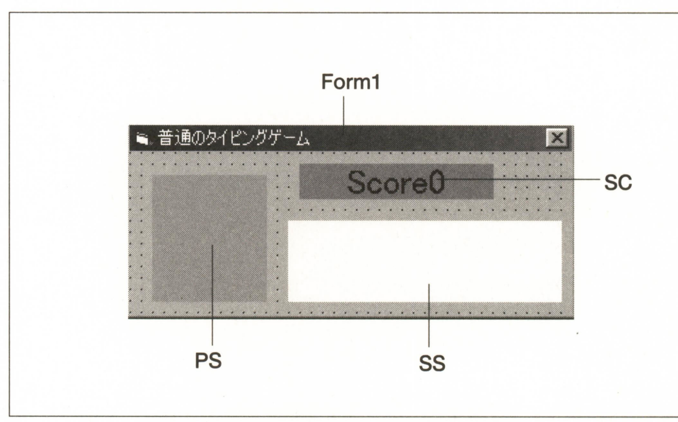
《写真１》これがProject1の画面



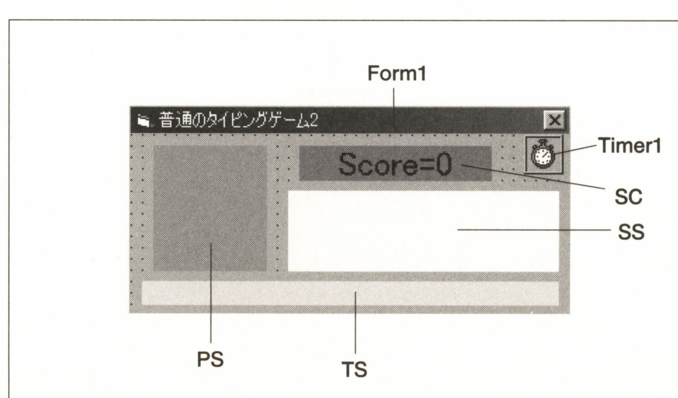
《写真２》タイムゲージがある Project2の画面



《写真３》Project2は終了後に得点表示される



Proiect1のフォーム配置図



Proiect2のフォーム配置図

★最近は梨ばかり食べています。技術がないので，皮をむくときは実を四つに割ってから。球体のままつるつるっとむけるようになりたいものです。（北海道札幌市・春陽19歳）……【影：つぐ美ちゃん，この梨の皮むいてよ。つぐ美：つるつるっと。影：実がない！】

《第１表》プロパティ表

Project1，Project2共通	
フォームの設定	
Name	Form1
BorderStyle	1 '固定(実線)
Caption	Project1：普通のタイピング・ゲーム Project2：普通のタイピング・ゲーム2
Font	ＭＳ Ｐゴシック
…Size	48
Height	2085
Left	0
LinkTopic	Form1
MaxButton	0 'False
MinButton	0 'False
ScaleHeight	1710
ScaleWidth	4680
StartUpPosition	2 '画面の中央
Top	0
Width	4770
ラベルの設定	
Name	SC
Alignment	2 '中央揃え
BackColor	&H00FF8080&
Caption	Score0
Font	ＭＳ Ｐゴシック
…Size	18
Height	375
Left	1800
TabIndex	2
Top	120
Width	2055
Name	SS
BackColor	&H00C0FFFF&
Height	855
Left	1680
TabIndex	1
Top	720
Width	2895
Name	PS
Alignment	2 '中央揃え
BackColor	&H008080FF&
Font	ＭＳ Ｐゴシック
…Size	72
Height	1335
Left	240
TabIndex	0
Top	240
Width	1215
Project2のみの設定	
ラベルの設定	
Name	TS
BackColor	&H00FFFFFF&
Height	255
Left	120
TabIndex	3
Top	1560
Width	4455
タイマーの設定	
Name	Timer1
Interval	1000
Left	4200
Top	0

《リスト1》Project1のプログラム・リスト

```
Dim Score As Integer

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    If Chr(KeyCode) = PS.Caption Then
        PS.Caption = Left(SS.Caption, 1)
        SS.Caption = Mid(SS.Caption, 2) + PF
        Score = Score + 5
    Else
        Score = Score - 1
    End If
    SC.Caption = "Score=" & Score
End Sub

Private Sub Form_Load()
    Randomize Timer
    PS.Caption = PF
    For i = 0 To 3
        SS.Caption = SS.Caption + PF
    Next

End Sub

Function PF() As String
    PF = Chr(Int(Rnd * 26) + 65)
End Function
```

《リスト2》Projext2のプログラム・リスト

```
Const TMax = 60
Dim Score As Integer, Tm As Integer

Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
    If Timer1.Enabled = False Then
        First
        Timer1.Enabled = True
        Exit Sub
    End If
    If Chr(KeyCode) = PS.Caption Then
        PS.Caption = Left(SS.Caption, 1)
        SS.Caption = Mid(SS.Caption, 2) + PF
        Score = Score + 5
    Else
        Score = Score - 1
    End If
    SC.Caption = "Score=" & Score
End Sub

Private Sub Form_Load()
    Randomize Timer
    Timer1.Enabled = False
End Sub

Function PF() As String
    PF = Chr(Int(Rnd * 26) + 65)
End Function

Private Sub Timer1_Timer()
    DS.Width = TS.Width / TMax * (TMax - Tm)
    Tm = Tm + 1
    If Tm > TMax Then
        Timer1.Enabled = False
        MsgBox "Score=" & Score
        Score = 0: Tm = 0
        PS.Caption = "": SS.Caption = ""
        DS.Width = TS.Width
    End If
End Sub

Sub First()
    PS.Caption = PF
    For i = 0 To 3
        SS.Caption = SS.Caption + PF
    Next
End Sub
```

CHECKER FLAG

つぐ美：本来，ベーマガへの投稿は１作品ごとが原則なのですが，リストが短いこともあり，特別に２本同時掲載です。

Dr.D：ゲーム内容は単純じゃが，このリストの短さはなかなかのものじゃ。ベーマガでは，できれば毎月１本はこういったリストの短い作品を紹介していきたいものじゃな。VBのフォームに用意されているKeyDownプロシージャをうまく利用しているのが，このリストの短さにつながっておるな。

編：この２本を見比べていくと，タイム制限のあるProject2のほうが，ゲームとして面白味が増していることがわかるハズです。こういった改造行程がわかる作品というのも，プログラム初心者の役に立つのではないでしょうか？

★AIBOに仲間ができましたが，まだまだ高いですね。姪っ子に「ロボットペットはやってるのよね」なんて言われ，猫型ロボットを買わされました。AIBOが数千円で買える時代はいつになるのでしょう。（秋田市・**おやじ工房36歳**）……【**つぐ美：**私もほしいなぁ。】

Windows(Visual Basic V6.0)用



同じ形式で暗号化はしないゼ

実用

RANDOM CODE Ver.1.03

（ランダム コード）

YUITA（18歳）

内容

テキストファイルをランダムに暗号化します。また，パスワードをつけて暗号化することで第三者によって内容を見られる心配もありません。

2001年8月号で指摘されたファイルサイズの制限をはずし，処理スピードを高めました。

使いかた

「ファイル選択」を押して暗号化，または解読したいファイルを選択します。パスワードを使わない場合はキャンセルを押してください。

解読する際に，パスワードがかかっているとパスワード入力画面になるので，パスワードを入力してください。暗号化，または解読が終了した後，保存先のファイルを指定してください。

暗号化のしくみ

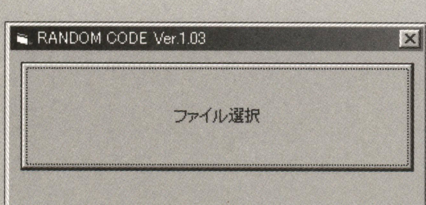
暗号化は文字の文字コードを1桁ずつ取り出して，10(100)の位の数をランダムに決めて再び文字に変換します。このとき，2バイト文字は文字コードが負になるため，1バイト文字と区別します。そのために1バイト文字には1の位が2で50以下の数を，2バイト文字には1の位が3で50以下の数を文字に変換し，それぞれの変換後の文字の後に付けます。

解読のしくみ

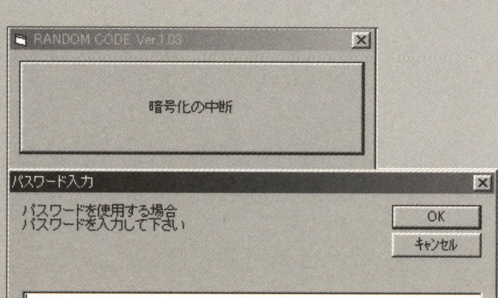
解読は文字の文字コードを取り出し，1の位の数だけを並べていきます。文字コードで，50以下で1の位が“2”，“3”の文字が出てきたら，そこまでが1文字分の文字コードとなります。その際，1の位が“2”の場合は1バイト文字，“3”の場合は2バイト文字となり，2バイト文字である場合は

《第1表》暗号＆解読の変換例

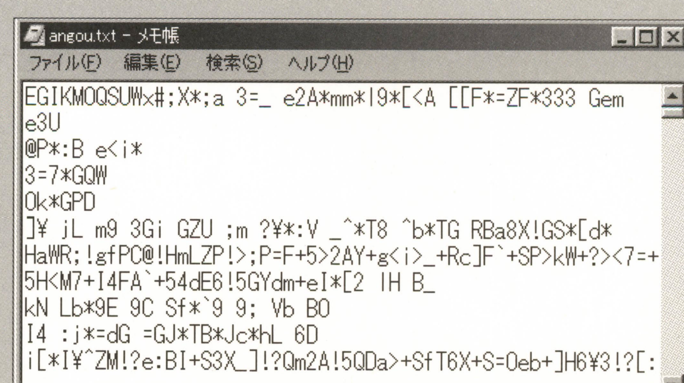
暗号化変換例
A → 65,42 → 6,5,42 → 76,85,42 → L,U,*→LU*
あ → -32096,43 → 32096,43 → 3,2,0,9,6,43 → 63,82,80,79,66,43 → ?,R,P,O,B,+ → ?RPOB+
解読変換例
LU* → L,U,* → 76,85,42 → 6,5 → 65 → A
?RPOB+ → ?,R,P,O,B,+ → 63,82,80,79,66,43 → 3,2,0,9,6 → 32096 → -32096 → あ



《写真1》ボタン一つで操作できる



《写真2》パスワードをかけることも可能



《《写真3》暗号化された文章は毎回異なるゾ

★地元のゲーム屋がつぶれました。別にいいけど。最近，何曜日かわからなくなってきた。（鹿児島県川内市・グリマー27歳）……
【影：10月号は失礼しました。市名をまちがえてしまいました。ごめんなさい。それにしても，ハガキがびっちり文字で埋まっています。】

CHECKER FLAG

Dr.D：乱数を使って暗号化する，というアイデアがいいのぉ。この方法は，いわばダミーの部分に乱数を加えるため，暗号化された文章は毎回異なるが，解析プログラムは，乱数の値に依存せず行えるというわけじゃ。

編：何回か言っていますが，一口CHECKER FLAGなどのコメントを参考にした再投稿があると，ありがたいですね。新しいコーナーを作ったかいがあるというものです。

つぐ美：ここ最近は，この暗号化ソフトの投稿が多いわよね～。正直言って，チェックしていて面白いと言えるジャンルではないため，毎月毎月送られてくると飽きてきてしまいます。どうしても暗号化ソフトを投稿したいなら，面白味のある独自性の機能がほしいところね。

影：逆に，いろんなソフトで暗号化された文章を解析するプログラムなら面白いんじゃないかな？

文字コードを負にしておきます。

パスワードのしくみ

パスワードを入力するときは，パスワードを暗号化する前の文章につけて一緒に暗号化します。そのあとでパスワードの入っていることを示すため，文字コードが120-129の文字と，文字コードが(パスワードの文字数+30)の文字を暗号化された文章の前に挿入します。

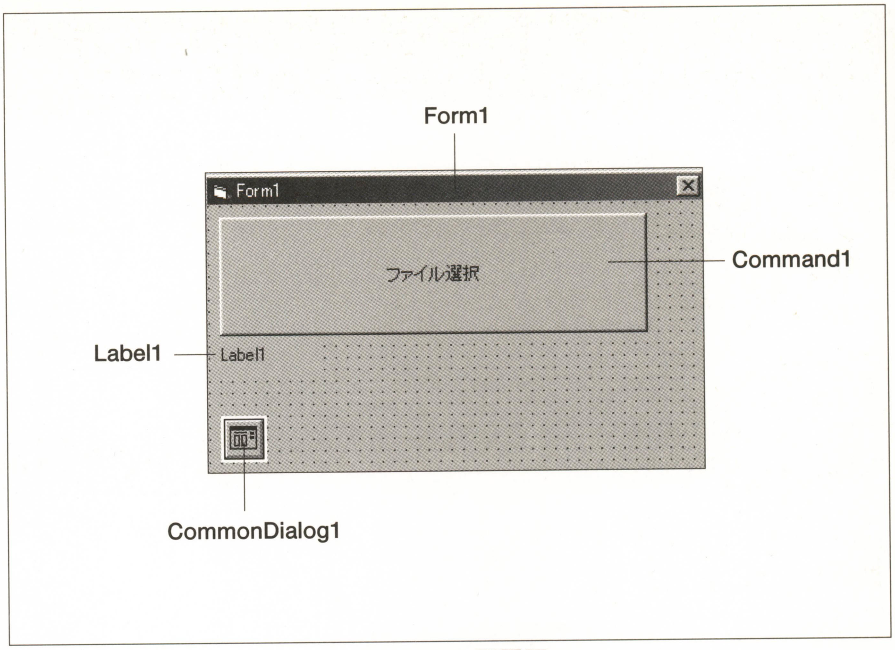
解読する際に，文字コードが120-129の文字を見つけた場合は，つぎの文字からパスワードの文字数がわかるので，その文字数分だけ解読してパスワードを取得しています。

暗号化されているかどうか

暗号化ランダムできめた30-110の数を利用して10個の文字列を作成し，暗号文の先頭と終わりに挿入します。ファイルが開かれた際にはまずここを調べるので，2度暗号化することはありません。

プログラムの入力法

コマンドボタン，ラベルをフォームに配置。「プロジェクト」メニューの「コンポーネント」を選択し「コンポーネント」ダイアログが現れるので「Microsoft Common Dialog Control 6.0」をチェックして，フォームに配置してください。



フォームの配置図

《第2表》プロパティ表

フォームの設定	
Name	Form1
AutoRedraw	-1 'True
BorderStyle	1 '固定(実線)
Caption	Form1
Height	2160
Left	0
LinkTopic	Form1
MaxButton	0 'False
MinButton	0 'False
ScaleHeight	1785
ScaleWidth	4680
StartUpPosition	2 '画面の中央
Top	0
Width	4770
コマンドボタンの設定	
Name	Command1
Caption	ファイル選択
Height	1215
Left	120
TabIndex	0
Top	120
Width	4455
ラベルの設定	
Name	Label1
Caption	Label1
Height	255
Left	120
TabIndex	1
Top	1440
Width	975
コモンダイアログの設定	
Name	CommonDialog1
CancelError	-1 'True
Filter	*.txt
Left	120
Top	2160

改造のアドバイス

暗号化すると元のファイルの数倍の大きさになってしまうので，圧縮してコンパクトにまとめるようにすると実用性が高まると思います。

苦労した点

処理スピードを高めるところが一番苦労しました。

前回投稿したものでも10K以上のファイルを暗号化することはできたのですが，あまりにもスピードが遅くなるので仕方なく制限することにしたのです。しかし，その原因が一つの変数に入力されているデータがあまりにも大きくなるため，処理が重たくなっているためだとわかったので，データを配列変数に分割して扱うようにしました。この改良で驚くほど処理スピードが上がり，ファイルサイズの制限もはずすことができました。

☆　☆　☆

★T「Kさん，プログラムの勉強しに海外行くんだって？！」　K「うん，Java（ジャワ）島にね」…地理の授業中に思いついたネタです。（愛媛県東宇和郡・**毛利祐允17歳**）……【**つぐ美：**よくできました。それだけ授業をまじめに聞いている証拠ね。**影：**そうかなぁ。】

『RANDOM CODE ver1.03』のプログラム・リスト

```
Option Explicit

Const this_title = "RANDOM CODE Ver.1.03"      'このソフトの名称
Dim msg As Integer                           'メッセージボックス
Dim fn As Integer                            'ファイルの空き番号
Dim ind As Long                              '変数のインデックス番号
Dim crlf As String                           '改行
Dim bun As Variant                           '暗号化されていないテキスト
Dim kaidokubun() As Variant                  '解読途中のテキスト
Dim angou() As Variant                       '暗号化途中のテキスト
Dim angoubun As Variant                      '暗号化されたテキスト
Dim temp As Variant                          '読み出し用データ
Dim moji_data As Variant                     '暗号文の数字データ
Dim shinkoudo As String                      '進行度
Dim moji_bite As String                      'moji_bite = 0 :1バイト文字
                                             'moji_bite = 1 :2バイト文字
Dim moji_code As String                      '文字コード
Dim fct As Integer                           'コマンドボタンの役割
Dim stp As Integer                           '作業中断
Dim pass_word As String                      'パスワード
Dim pass_word_length As Long                 'パスワードの文字数
Private Sub kaidoku() '解読
On Error GoTo error
Dim A, i, j As Long
Dim mojisuu As Long 'パスワードの文字数
ReDim kaidokubun(0)
bun = ""
ind = 0
moji_data = 0
moji_code = 0
Err = 1
'***文字を数値に変換→数値を文字へ変換***
For A = 1 To Len(angoubun)
   If A + 1 <= Len(angoubun) Then '改行を無視
     Do Until Mid(angoubun, A, 2) <> Chr(&HD) + Chr(&HA)
        A = A + 2
        If A >= Len(angoubun) Then
           Exit For
        End If
     Loop
   End If
   moji_code = Asc(Mid(angoubun, A, 1))
   If Int(moji_code / 10) <= 4 Then
      If Right(moji_code, 1) = 2 Then '1バイト文字
         kaidokubun(ind) = kaidokubun(ind) & Chr(moji_data)
         moji_data = ""
         If pass_word_length > 0 Then mojisuu = mojisuu + 1
      ElseIf Right(moji_code, 1) = 3 Then '2バイト文字
         kaidokubun(ind) = kaidokubun(ind) & Chr(-(moji_data))
         moji_data = ""
         If pass_word_length > 0 Then mojisuu = mojisuu + 1
      End If
      If Len(kaidokubun(ind)) > 500 Then '変数内をスマートに
         ind = ind + 1
         ReDim Preserve kaidokubun(ind)
      End If
      If pass_word_length > 0 And mojisuu = pass_word_length Then 'パスワード解読
         angoubun = Mid(angoubun, A + 1)
         Exit For
      End If
   Else
      moji_data = moji_data & Right(moji_code, 1)
   End If
   '***進行度の表示***
   If A Mod 100 = 0 Then
      shinkoudo = A / Len(angoubun) '進行度計算
      Form1.Line (Command1.Left, Command1.Top + Command1.Height)-(Command1.Left +
Command1.Width, Command1.Top + Command1.Height + 100), RGB(0, 255, 0), BF
      Form1.Line (Command1.Left, Command1.Top + Command1.Height)-(Command1.Left +
```

```
Command1.Width * shinkoudo, Command1.Top + Command1.Height + 100), RGB(255, 0, 0), BF
      Label1.Caption = Int(shinkoudo * 10000) / 100 & "%"
      If stp = 1 Then '作業の中断
         Err = 3
         GoTo error
      End If
      DoEvents
   End If
Next A
For A = 0 To ind '結合
   bun = bun & kaidokubun(A)
   kaidokubun(A) = ""
Next A
'***パスワードの照合***
If pass_word_length > 0 Then
   'パスワードあり
   If pass_word = bun Then
      pass_word_length = 0
      Call kaidoku
      Exit Sub
   Else
      Err = 2 'パスワードが違う
      GoTo error
   End If
Else
   'パスワードなし
   Form1.Cls
   fct = 0
   stp = 0
   Command1.Caption = "ファイルの選択"
   Label1.Caption = ""
   '保存
   fn = FreeFile
   CommonDialog1.DialogTitle = "解読データの保存先を指定して下さい"
   CommonDialog1.ShowSave
   Open CommonDialog1.FileName For Output As #fn
   Print #fn, bun
   Close #fn
   Exit Sub
End If
error: 'エラー処理
If Err = 1 Then
   msg = MsgBox("解読に失敗しました", , this_title)
ElseIf Err = 2 Then
   msg = MsgBox("パスワードが違います", , this_title)
ElseIf Err = 3 Then
   msg = MsgBox("解読を中断しました", , this_title)
ElseIf Err = 32755 Then 'ダイアログボックスでキャンセルボタンが押された
   msg = MsgBox("保存しませんでした", , this_title)
Else
   msg = MsgBox("解読に失敗しました", , this_title)
End If
Form1.Cls
fct = 0
stp = 0
Command1.Caption = "ファイルの選択"
Label1.Caption = ""
End Sub



Private Sub angouka() '暗号化
On Error GoTo error
Dim A, B, C As Long
ind = 0
moji_bite = 0
moji_code = 0
angoubun = ""
Err = 1
'***パスワードを入力***
pass_word = InputBox("パスワードを使用する場合" & crlf & "パスワードを入力して下さい", "パスワ
```

★FF映画の予告を見るだけでも，もう役者なんていらないのでは？とも思える…が，とりあえず「全日大ヒット！」（愛知県名古屋市・ズオウ・リアキ29歳）……【影：全日本プロレスとしては，三冠王座の早期奪回は至上命令だな。つぐ美：うーん，その全日かなー。】

```
ード入力")
If pass_word <> "" Then
   bun = pass_word & bun
End If
'***暗号化***
ReDim angou(Int(Len(bun) / 500))
For A = 1 To Len(bun)
    If A > (ind + 1) * 500 Then ind = ind + 1 '変数内をスマートに
       moji_bite = 0
       moji_code = Asc(Mid(bun, A, 1))
       If moji_code < 0 Then '２バイト文字
          moji_code = -moji_code
          moji_bite = 1
       End If
    For B = 1 To Len(moji_code)
       angou(ind) = angou(ind) & Chr((Int(Rnd * 6) + 5) * 10 + Mid(moji_code, B, 1))
    Next B
    If moji_bite = 0 Then '１バイト文字 OR ２バイト文字
       angou(ind) = angou(ind) & Chr((Int(Rnd * 2) + 3) * 10 + 2)
    Else
       angou(ind) = angou(ind) & Chr((Int(Rnd * 2) + 3) * 10 + 3)
    End If
    If A Mod (Int(Rnd * 10) + 10) = 0 Then 'ランダムで改行
       angou(ind) = angou(ind) & crlf
    End If
    '***進行度の表示***
    If A Mod 100 = 0 Then
       shinkoudo = A / Len(bun) '進行度計算
       Form1.Line (Command1.Left, Command1.Top + Command1.Height)-(Command1.Left +
Command1.Width, Command1.Top + Command1.Height + 100), RGB(0, 255, 0), BF
       Form1.Line (Command1.Left, Command1.Top + Command1.Height)-(Command1.Left +
Command1.Width * shinkoudo, Command1.Top + Command1.Height + 100), RGB(255, 0, 0), BF
       Label1.Caption = Int(shinkoudo * 10000) / 100 & "%"
       If stp = 1 Then '作業の中断
          Err = 2
          GoTo error
       End If
       DoEvents
    End If
Next A
For A = 0 To ind '結合
    angoubun = angoubun & angou(A)
    angou(A) = ""
Next A
Form1.Cls
fct = 0
stp = 0
Command1.Caption = "ファイルの選択"
Label1.Caption = ""
'***パスワードについてのデータ入力***
If pass_word <> "" Then
   angoubun = Chr(Int(Rnd * 10) + 120) & Chr(Len(pass_word) + 30) & angoubun
End If
'***暗号化された証***
Dim hanbetsu_code As String
Dim hanbetsu As String
hanbetsu_code = (Int(Rnd * 80) + 30)
For C = 0 To 9
hanbetsu = hanbetsu & Chr(hanbetsu_code + C * 2)
Next C
angoubun = hanbetsu & angoubun '先頭につける
hanbetsu = ""
hanbetsu_code = (Int(Rnd * 80) + 30)
For C = 0 To 9
hanbetsu = hanbetsu & Chr(hanbetsu_code + C * 2)
Next C
angoubun = angoubun & hanbetsu '終わりにつける
'保存
fn = FreeFile
CommonDialog1.DialogTitle = "暗号データの保存先を指定して下さい"
CommonDialog1.ShowSave
```

```
Open CommonDialog1.FileName For Output As #fn
Print #fn, angoubun
Close #fn
Exit Sub
error: 'エラー処理
If Err = 1 Then
   msg = MsgBox("暗号化に失敗しました", , this_title)
ElseIf Err = 2 Then
   msg = MsgBox("暗号化を中止しました", , this_title)
ElseIf Err = 32755 Then 'ダイアログボックスでキャンセルボタンが押された
   msg = MsgBox("保存しませんでした", , this_title)
Else
   msg = MsgBox("暗号化に失敗しました", , this_title)
End If
Form1.Cls
fct = 0
stp = 0
Command1.Caption = "ファイルの選択"
Label1.Caption = ""
End Sub


Private Sub Command1_Click()
On Error GoTo error
'***ファイル選択***
If fct = 0 Then
   fn = FreeFile
   CommonDialog1.DialogTitle = "ファイルを選択して下さい"
   CommonDialog1.ShowOpen
   Open CommonDialog1.FileName For Input As #fn
       temp = StrConv(InputB(LOF(fn), #fn), vbUnicode)
   Close #fn
   Call shikibetsu
'***作業中断***
Else
   stp = 1
   Command1.Caption = "ファイルの選択"
End If
Exit Sub
error: 'エラー処理
If Err = 32755 Then  'ダイアログボックスでキャンセルボタンが押された
   msg = MsgBox("開きませんでした", , this_title)
ElseIf Err = 62 Then '扱えないデータを指定した
   msg = MsgBox("扱えないファイルが指定されました", , this_title)
ElseIf Err = 53 Then 'ファイルが見つからない
   msg = MsgBox("ファイルが見つかりません", , this_title)
Else 'その他のエラー
   msg = MsgBox("失敗しました", , this_title)
End If
End Sub


Private Sub Form_Load()
Randomize
crlf = Chr(&HD) + Chr(&HA) '改行
fct = 0
stp = 0
Form1.Caption = this_title
Form1.AutoRedraw = True
Label1.Caption = ""
Label1.Top = Command1.Top + Command1.Height + 120
Label1.Left = Command1.Left
End Sub

Private Sub shikibetsu() 'テキストが暗号化されているかを識別
On Error GoTo error
Dim A, B As Long
'***暗号化されているかを調べる***
If Len(temp) > 20 And (Asc(Mid(temp, 1, 1)) >= 30) And (Asc(Mid(temp, 1, 1)) < 110) Then
    For A = 1 To 9
        If Asc(Mid(temp, A + 1, 1)) = Asc(Mid(temp, 1, 1)) + A * 2 And Asc(Mid(temp, Len(temp)
```

★Xboxの日本での発売日が，じつは2222年2月22日だったらどうしよう。（群馬県前橋市・トランス19歳）……【**編：**そんな未来にはXboxも忘れ去られ，その世代にマッチしたマシンが登場しているでしょう。Xboxの日本発売日は2002年2月22日です。念のため。】

```
- 11 + A, 1)) = Asc(Mid(temp, Len(temp) - 11, 1)) + A * 2 Then
            B = B + 1
        Else
            Exit For
        End If
    Next A
    If B = 9 Then '暗号化されている
        Command1.Caption = "解読の中断"
        fct = 1
        temp = Mid(temp, 11, Len(temp) - 12)
        angoubun = temp
        '***パスワードが入力されているか***
        If (Asc(Mid(angoubun, 1, 1)) >= 120) And (Asc(Mid(angoubun, 1, 1)) <= 130) Then
            pass_word_length = Asc(Mid(angoubun, 2, 1)) - 30
            angoubun = Mid(angoubun, 3)
            pass_word = InputBox("パスワードがかかっているので" & crlf & "パスワードを入力して下さい", "パスワード入力")
        End If
        Call kaidoku '解読
    Else '暗号化されていない
        GoTo ango '暗号化
    End If
Else
ango: '暗号化されていない
    Command1.Caption = "暗号化の中断"
    fct = 2
    bun = temp
    Call angouka '暗号化
End If
Exit Sub
error: 'エラー処理
msg = MsgBox("このファイルは扱えません", , this_title)
End Sub
```

ゲリラコラム（その13）

「VBって凄いや」

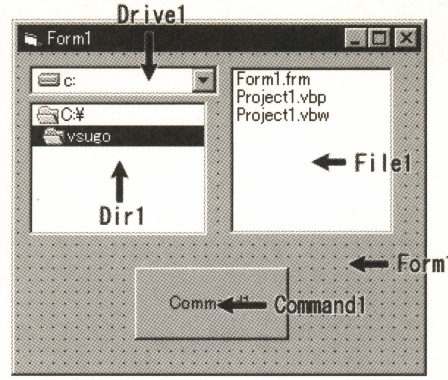
松原拓也

VisualBasicのちょっとイイ話を紹介する「VBって凄いや」です。

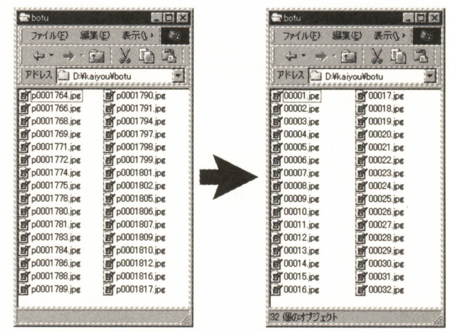
さて，僕の使っているデジカメは，撮影するとファイル名が勝手に付くのですが，たとえば“p0002195.jpg”というふうに長い名前になってしまったりします。ファイル名の変更は何十個もあると大変です。そこで，一気にリネームしてしまう「ファイル名一括変更プログラム」をVBで作ってみましょう。

(約15分ほど格闘)

……ハイ，できました。ボタンを押すとこのとおり，連続した番号にリネームされます！なお，このプログラムは勝手にファイル名を変えるという過激なものですので，扱いには十分注意してください。



▲フォームのレイアウト



▲実行結果（右）。ボタン1発でファイル名が連番に。便利！

《リスト》ファイル名一括変更プログラム（VisualBasic）

```
Private Sub Command1_Click()
    Dim iNum As Integer 'ファイル名にする番号
    Dim iID As Integer 'インデックス番号
    Dim strSrc As String '変更前のファイル名
    Dim strDst As String '変更後のファイル名
    Dim fullpath As String 'パス名

    Command1.Enabled = False
    fullpath = Dir1.Path
    If Right$(fullpath, 1) <> "¥" Then
        fullpath = fullpath + "¥"
    End If

    iNum = 1 'スタート番号
    For iID = 0 To File1.ListCount - 1
        File1.ListIndex = iID
        strSrc = File1.filename
        strDst = Format$(iNum, "00000")  'ファイル名
        strDst = strDst + Mid$(strSrc, InStr(strSrc, "."), 10) '拡張子
        strSrc = fullpath + strSrc
        strDst = fullpath + strDst
        Name strSrc As strDst 'リネーム
        iNum = iNum + 1
    Next
    Command1.Enabled = True
    File1.Refresh
End Sub

Private Sub Dir1_Change()
    File1.Path = Dir1.Path
End Sub

Private Sub Drive1_Change()
    Dir1.Path = Drive1.Drive
End Sub

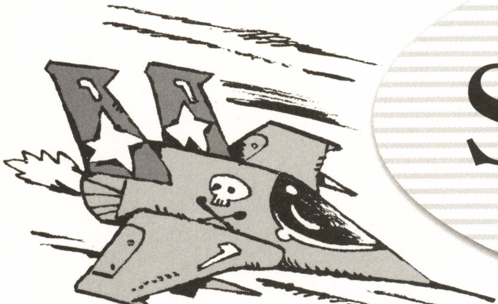
Private Sub Form_Load()
    File1.Pattern = "*.jpg;*.gif;*.jpeg"  '対象を画像に限定
End Sub
```


★やっとHP本格運用開始！www6.ocn.ne.jp/~watasino/　記念イベント（？）開始しました。（島根県簸川郡・私野市長24歳）……【編：さっそくHP見ましたよ。記念イベントって，ベーマガクイズのこと？影：応募しておいたので，プレゼントちょうだいね。よろしく！】

Windows(Visual Basic V5.0)用

“Shino”を倒しまくれ

Shino Buster（シノ バスター）
BM Version（ベーマガ バージョン）



てぃこく（22歳）

ゲームの内容

オーソドックスなシューティング・ゲームです。急に作りたくなったので作ってみました。

ストーリー

戦闘機が地球に帰還中，妙な領域に突入してしまった。そこで大量に現れる物体“Shino”を撃ち落としつづけ，地球に帰ることにした。しかし，新たに判明した未確認物体として「影さんの顔」と呼ばれるものが現れ，戦闘機の行く手を阻む。果たして無事に帰還できるか!?

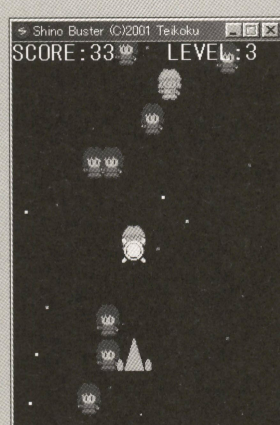
入力方法

(1)VBを起動し，フォームにオブジェクトを貼り付けます。
(2)フォームのリスト，モジュールのリストをそれぞれ入力します。
(3)VBのメニューから「プロジェクト」→「Project1のプロパティ」を選択し，「全般」の「スタートアップの設定」を「Sub Main」に変更して，保存します。
(4)一度VBを終了し，再び起動し，読み込んで実行してください。

なお，キャラクター画像“Chip.bmp”は横256×縦32で，一つのキャラが32×32になっています。順番は戦闘機，レーザー，Shino（青），Shino（灰），影さん



《写真1》スタンダードなシューティング



《写真2》灰色の敵はいくら弾を当てても倒せない



《写真3》影さんの顔は、本人同様かなりやっかい



《写真4》高得点を目指す遊びかたをしよう

CHECKER FLAG

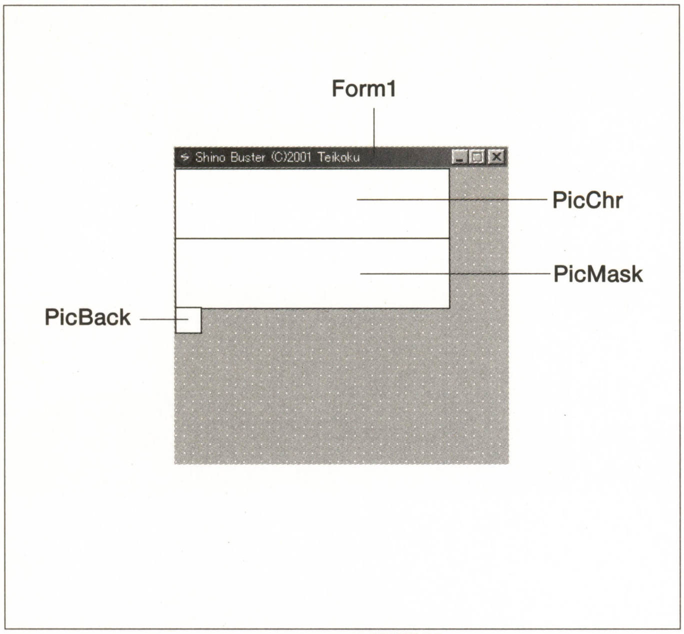
つぐ美：ホントにオーソドックスなシューティングね。ゲームとしては正直言ってすぐに飽きてしまうのですが，プログラム初心者にとっては，こういう単純なゲームのリストは重宝しますよね。

Dr.D：敵の動きなんかも乱数を使用した単純な形式じゃ。改造するのであれば，敵の動きじゃな。ホーミングや移動速度の増減など，いろいろ試してみてはどうかな？

編：よくベーマガＨＰの掲示板で「〜のゲームの作りかたがわからない。教えて」といった質問がありますが，そういう人は，まず自分で作成する前に，こういった単純な形式のリストの研究や改造から始めるのがオススメです。人から教えてもらうよりも勉強になるハズですよ。

影：せっかくベーマガには，数多くの投稿プログラムのリストが掲載されるのだから，それを利用しない手はないゾ。遊ぶだけでなく，プログラムの勉強にも投稿プログラムは役に立つのだ。

★残暑きびしいおり，財布の中だけ異様に寒い！（茨城県古河市・**年齢不相応41歳**）……【**影：**私なんて，年がら年中，財布の中はお寒いのでござる。**編：**何に使ってるのか不思議なほど，いつもお金ないですよね。貯金もなさそうだし。**影：**そこまで言わなくたって。】



フォームの配置図

《第１表》変数表

スプライトの内容					
変数	内容	X	Y	Data1	Data2
l(n)	レーザー	X座標	Y座標	レーザーの状態	
e(n)	敵	X座標	Y座標	敵の状態	X方向の移動量
s(n)	背景の星	X座標	Y座標	星の色	

スプライト以外の変数	
mx	自機のX座標
my	自機のY座標
FieldX	画面の幅
FieldY	画面の高さ
flag	ゲームの状態
HiScore	ハイスコア
Score	スコア
Lmax	レーザーの最大数
Emax	敵の最大数
Level	レベル
hx、hy	当たり判定用
vx、vy	自機移動用
i、j	ループ用
その他	汎用



chip-m.bmpの画像



chip.bmpの画像

の顔，二つ空けてから爆発パターンです。

“Chip_m.bmp”は“Chip.bmp”の抜き型となっています。

また，爆発音用WAVEファイル"bomb.wav"をプログラムと同じフォルダに入れておくと，破壊したときに音が鳴るようになります。

遊びかた

最初にタイトル画面が表示されます。スペース・キーでゲームが始まります。

操作方法は，カーソル・キーで自機(灰色の三角形型戦闘機）を動かし，スペース・キーでレーザーを発射します。レーザーは一画面に５発までしか撃てません。できるだけ多くの飛行物体“Shino”を破壊することがこのゲームの目的です。点数を稼ぐと，レベルが上がり，敵の数が増えていきます。レベルは最高20です。

飛行物体は青と灰色の２種類があります。このうち青は１発で破壊可能ですが，灰色はたとえ256発当てても破壊することはできません。

影さんの顔はレベル３から登場します。何発当てても破壊できないうえ，移動速度が速いです。

自機が敵にぶつかるとゲーム・オーバー。しばらくするとタイトル画面に戻ります。

ゲームを終了するにはタイトル画面でESCキーを押すか，ウィンドウの右上にある×ボタンを押してください。

《第１表》プロパティ表

フォームの設定	
Name	Form1
AutoRedraw	-1 'True
BorderStyle	1 '固定(実線)
Caption	Shino Buster (C)2001 Teikoku
Font	ＭＳ ゴシック
ForeColor	&H00FFFFFF&
Height	4500
Icon	(アイコン)
Left	0
LinkTopic	Form1
MaxButton	0 'False
ScaleHeight	275
ScaleMode	3 'ピクセル
ScaleWidth	312
StartUpPosition	3 'Windows の既定値
Top	0
Width	4770

ピクチャボックスの設定	
Name	PicBack
Appearance	0 'フラット
AutoRedraw	-1 'True
BackColor	&H80000005&
Font	ＭＳ Ｐゴシック
ForeColor	&H80000008&
Height	25
Left	0
ScaleHeight	23
ScaleMode	3 'ピクセル
ScaleWidth	23
TabIndex	2
Top	128
Visible	0 'False
Width	25

Name	PicMask
Appearance	0 'フラット
AutoRedraw	-1 'True
BackColor	&H80000005&
Font	ＭＳ Ｐゴシック
ForeColor	&H80000008&
Height	66
Left	0
ScaleHeight	64
ScaleMode	3 'ピクセル
ScaleWidth	256
TabIndex	1
Top	64
Visible	0 'False
Width	258

Name	PicChr
Appearance	0 'フラット
AutoRedraw	-1 'True
BackColor	&H80000005&
Font	ＭＳ Ｐゴシック
ForeColor	&H80000008&
Height	66
Left	0
ScaleHeight	64
ScaleMode	3 'ピクセル
ScaleWidth	'256
TabIndex	0
Top	0
Visible	0 'False
Width	258

改造のポイント

グラフィックを変えてみると違った感じになります。BGMを鳴らしてみるといいかもしれません。

スピードを調整したい人はGameMainの後半にある“Delay 50”の50を変えることで調整ができます（大きくすると遅くなります)。

同じ部分にある“Emax = Level * 5 - 1”は，敵の最大出現数を算出する式になっていますので，ここを変更することで敵の数を調整できます（ただし99以下に抑えてください)。

MoveEnemyとPrint_Srcにある“i Mod 6”の6を変えることで，灰色のShinoの出現頻度を調整することができます。

《リスト１》フォームのプログラム・リスト

```
Private Sub Form_Unload(Cancel As Integer)
  End
End Sub
```


★はじめて愛読者カードを送ります。そして突然ですが，ソースコードを読んでいると眠くなるのはやはり気合が足りないせいでしょうか。(東京都足立区・やん22歳）……【影：グー。編：プログラムチェック中に寝ないでください。影：催眠プログラムなんだよ。】

《リスト２》モジュールのプログラム・リスト

```
' BitBltなどの命令が使えるように宣言
Declare Function BitBlt Lib "gdi32" (ByVal hDestDC As Long, ByVal X As Long, ByVal Y As Long,
ByVal nWidth As Long, ByVal nHeight As Long, ByVal hSrcDC As Long, ByVal Xsrc As Long, ByVal
Ysrc As Long, ByVal dwRop As Long) As Long
Declare Function GetTickCount Lib "kernel32" () As Long
Declare Function sndPlaySound Lib "winmm.dll" Alias "sndPlaySoundA" (ByVal lpszSoundName As
String, ByVal uFlags As Long) As Long
Declare Function GetAsyncKeyState Lib "user32" (ByVal vKey As Long) As Integer
Declare Function StretchBlt Lib "gdi32" (ByVal hDC As Long, ByVal X As Long, ByVal Y As Long,
ByVal nWidth As Long, ByVal nHeight As Long, ByVal hSrcDC As Long, ByVal Xsrc As Long, ByVal
Ysrc As Long, ByVal nSrcWidth As Long, ByVal nSrcHeight As Long, ByVal dwRop As Long) As Long
' 定数の宣言
Public Const SRCCOPY = &HCC0020
Public Const SRCINVERT = &H660046
Public Const SRCPAINT = &HEE0086
Public Const SRCAND = &H8800C6
' 自作データ型:Sprite型の定義
Public Type Sprite
  X As Long      ' X座標
  Y As Long      ' Y座標
  Data1 As Long ' データ1
  Data2 As Long ' データ2
End Type

Public mx As Integer, my As Integer, flag As Integer, HiScore As Long
Public l(4) As Sprite, e(99) As Sprite, s(39) As Sprite
Public Score As Long, Lmax As Integer, Emax As Integer, Level As Integer
Public Const FieldX = 256  ' 画面の幅
Public Const FieldY = 384  ' 画面の高さ

'このプロジェクトのスタート地点
Sub Main()
  Randomize
  Form1.PicBack.Move 0, 0, FieldX, FieldY
  Form1.PicChr.Move 0, 0, 256, 32
  Form1.PicMask.Move 0, 0, 256, 32
  Form1.PicChr.Picture = LoadPicture(MyForm & "chip.bmp")    ' キャラ画像読み込み
  Form1.PicMask.Picture = LoadPicture(MyForm & "chip_m.bmp") ' マスク画像読み込み
  Form1.Width = FieldX * Screen.TwipsPerPixelX
  Form1.Height = FieldY * Screen.TwipsPerPixelY
  Form1.Show
  HiScore = 100
  GameMain  ' ゲームのメインルーチンへ
End Sub

Sub GameMain()
Dim i As Integer
Start:
  GameTitle  ' タイトル画面へ
  Delay 100
  ' ↓スペースキーとESCの両方とも押されていないか確かめる
  While (GetAsyncKeyState(32) <> 0 Or GetAsyncKeyState(27) <> 0): Wend
  '  決戦前夜 (ゲーム開始前の初期設定)
  Form1.ForeColor = QBColor(15):  Form1.FontSize = 16
  Lmax = 4  ' 画面に出るレーザーの最大数
  ' レーザーを発射していない状態にする
  For i = 0 To Lmax
    l(i).Data1 = 0
  Next i
  ' 敵の位置や動く方向、その移動量を設定する
  For i = 0 To 99
    e(i).X = Int(Rnd * 27) * 8: e(i).Y = -i * 32 - 160: e(i).Data1 = 0
    e(i).Data2 = Int(Rnd * 5) - 2
  Next i
  ' 背景の星の位置や色を設定する
  For i = 0 To 39
    s(i).X = Int(Rnd * 240): s(i).Y = Int(Rnd * 384) - 64: s(i).Data1 = Int(Rnd * 7) + 1
  Next i
  ' 自機の位置などを設定
  mx = 4 * 32: my = 10 * 32: Score = 0: Level = 1: flag = 0
  ' ↓ゲームオーバーでない(flag=0)限り、このループは繰り返される
  While (flag = 0)
    Emax = Level * 5 - 1
    MoveShip  ' 自機移動
    Laser     ' レーザー関係
    MoveEnemy ' 敵移動
    MoveStar  ' 星移動
    Delay 50  ' 時間待ち
    Print_Src ' 画面に表示
  Wend
  'ゲームオーバー
  PlayWAV MyForm & "bomb.wav", 1
  Delay 100
  Form1.ForeColor = QBColor(10):  Form1.FontSize = 20
  Locate 60, 140: Form1.Print "GAME OVER"
  If Score > HiScore Then HiScore = Score
  Delay 5000
  GoTo Start
End Sub

' 自機移動
Sub MoveShip()
Dim vx As Integer, vy As Integer, i As Integer
  vx = (GetAsyncKeyState(37) <> 0) - (GetAsyncKeyState(39) <> 0)  ' 左右の移動
  vy = (GetAsyncKeyState(38) <> 0) - (GetAsyncKeyState(40) <> 0)  ' 上下の移動
  If (mx + vx * 8) < 0 Or (mx + vx * 8) >= FieldX - 32 Then vx = 0
  If (my + vy * 8) < 0 Or (my + vy * 8) >= FieldY - 48 Then vy = 0
  mx = mx + vx * 8: my = my + vy * 8
  ' ↓スペースキーを押したかどうか
  If GetAsyncKeyState(32) <> 0 Then
    ' レーザー発射のチェック
    For i = 0 To Lmax
      If l(i).Data1 = 0 Then
        l(i).Data1 = 1: l(i).X = mx: l(i).Y = my - 16: Exit For
      End If
    Next i
  End If
End Sub

' レーザー移動
Sub Laser()
Dim i As Integer
  For i = 0 To Lmax
    If l(i).Data1 = 1 Then
      l(i).Y = l(i).Y - 16
      If l(i).Y <= -32 Then l(i).Data1 = 0 ' 画面の上に消えたかどうか
    End If
    If l(i).Data1 = 2 Then l(i).Data1 = 0
 Next i
End Sub

' 敵移動
Sub MoveEnemy()
Dim i As Integer, j As Integer, hx As Integer, hy As Integer
  ' 敵番号を6で割った時の余りが5ならば、灰色のShino
  ' 敵番号を11で割った時の余りが10ならば、影さん
  ' それ以外は青のShino
  For i = 0 To Emax
    e(i).Y = e(i).Y + 8: e(i).X = e(i).X + e(i).Data2 * 2
    If (i Mod 11 = 10) Then e(i).Y = e(i).Y + 8
    If e(i).X < 0 Or e(i).X > FieldX - 40 Then e(i).Data2 = -e(i).Data2
    If e(i).Y >= FieldY Or e(i).Data1 = 1 Then
      e(i).X = Int(Rnd * 27) * 8: e(i).Y = -Int(Rnd * 7) * 16 - 160: e(i).Data1 = 0
      e(i).Data2 = Int(Rnd * 5) - 2
    Else
      ' 敵とレーザーの当たり判定
      For j = 0 To Lmax
        If l(j).Data1 = 1 Then
          hx = l(j).X - e(i).X: hy = l(j).Y - e(i).Y
          If hx > -16 And hx < 12 And hy > -24 And hy < 24 Then
            If (i Mod 6 <> 5) And (i Mod 11 <> 10) Then
              e(i).Data1 = 1: l(j).Data1 = 0
              PlayWAV MyForm & "bomb.wav", 1
              Score = Score + 1
              If Score >= (Level * (Level + 1)) * 5 Then Level = Level + 1
              If Level > 20 Then Level = 20
            Else
              l(j).Data1 = 2
            End If
          End If
```

★今年の夏は「しろくま」というアイスに大ハマリ。ちょっと高いけどね。練乳が全体にかかっていて，その上にあずき，果物（桃・梨）ゼリー(？)が載ってる氷菓。鹿児島名物。（愛知県知多郡・**四谷一獲26歳**）……**【つぐ美：鹿児島在住の方，ぜひ贈ってきて〜。】**

```
      End If
    Next j
    If e(i).Data1 = 0 Then
      ' 敵と自機の当たり判定
      hx = mx - e(i).X: hy = my - e(i).Y
      If hx > -20 And hx < 16 And hy > -20 And hy < 16 And flag = 0 Then flag = 1
    End If
  End If
Next i
End Sub

' 背景の星移動
Sub MoveStar()
Dim i As Integer
  For i = 0 To 39
    s(i).Y = s(i).Y - (i >= 20) * 12 + 4
    If s(i).Y > FieldY Then
      s(i).X = Int(Rnd * 240): s(i).Y = Int(Rnd * 384) - 384: s(i).Data1 = Int(Rnd * 7) + 1
    End If
  Next i
End Sub

' 画面に表示
Sub Print_Src()
Dim i As Integer, j As Integer
  Form1.PicBack.Line (0, 0)-(FieldX, FieldY), QBColor(0), BF
  ' まずは背景の星（遠く）
  For i = 0 To 19
    Form1.PicBack.Line (s(i).X+1, s(i).Y+1)-(s(i).X+2, s(i).Y+2), QBColor(s(i).Data1), BF
  Next i
  ' 次に背景の星（近く）
  For i = 20 To 39
    Form1.PicBack.Line (s(i).X+0, s(i).Y+0)-(s(i).X+2, s(i).Y+2), QBColor(s(i).Data1+8), BF
  Next i
  ' そして敵の表示
  For i = 0 To Emax
    If e(i).Y >= -32 Then
      If e(i).Data1 = 1 Then
        j = 224
      ElseIf (i Mod 6) = 5 Then
        j = 96
      ElseIf (i Mod 11) = 10 Then
        j = 128
      Else
        j = 64
      End If
      BmpKasane Form1.PicChr.hDC,Form1.PicMask.hDC,j,0,32,32,Form1.PicBack.hDC,e(i).X,e(i).Y
    End If
  Next i
  ' さらにレーザーの表示
  For i = 0 To Lmax
    If l(i).Data1 > 0 Then
      BmpKasane Form1.PicChr.hDC, Form1.PicMask.hDC, -(l(i).Data1 = 2) * 192 + 32, 0, 32, 32,
Form1.PicBack.hDC, l(i).X, l(i).Y
    End If
  Next i
  ' それから自機の表示
  If flag = 0 Then
    BmpKasane Form1.PicChr.hDC, Form1.PicMask.hDC, 0, 0, 32, 32, Form1.PicBack.hDC, mx, my
  Else
    BmpKasane Form1.PicChr.hDC, Form1.PicMask.hDC, 224, 0, 32, 32, Form1.PicBack.hDC, mx, my
  End If
  ' 最後にスコアとレベルの表示
  Form1.PicBack.Refresh
  BmpCopy Form1.PicBack.hDC, 0, 0, FieldX, FieldY, Form1.hDC, 0, 0
  Locate 0, 0: Form1.Print "SCORE:" & Score
  Locate 144, 0: Form1.Print "LEVEL:" & Level
End Sub

' タイトル画面
Sub GameTitle()
  Form1.Line (0, 0)-(FieldX, FieldY), QBColor(0), BF
  Form1.ForeColor = QBColor(15): Form1.FontSize = 12
  Locate 60, 20: Form1.Print "[PROJECT SS-98]"
  Form1.ForeColor = QBColor(14): Form1.FontSize = 40
  Locate 20, 50: Form1.Print "Shino"
  Locate 60, 100: Form1.Print "Buster"
  Form1.ForeColor = QBColor(15): Form1.FontSize = 16
  Locate 70, 160: Form1.Print "BM Version"
  Form1.ForeColor = QBColor(15): Form1.FontSize = 12
  Locate 20, 250: Form1.Print "Push Space Key To Start"
  Locate 35, 270: Form1.Print "Push ESC Key To Exit"
  Form1.ForeColor = QBColor(11): Form1.FontSize = 16
  Locate 40, 200: Form1.Print "Best Score:" & HiScore
  Form1.ForeColor = QBColor(11): Form1.FontSize = 12
  Locate 60, 310: Form1.Print "(C)2001 Teikoku"
  Locate 40, 330: Form1.Print "All Rights Reserved."
  Form1.Refresh
  Delay 100
  While (GetAsyncKeyState(32) <> 0 Or GetAsyncKeyState(27) <> 0): Wend
  flag = 0
  While (flag = 0)
    If GetAsyncKeyState(32) <> 0 Then flag = 1
    If GetAsyncKeyState(27) <> 0 Then flag = 2
    Delay 50
  Wend
  If flag = 2 Then End
End Sub

' 文字の表示位置指定
Sub Locate(ByVal X As Integer, Y As Integer)
  Form1.CurrentX = X: Form1.CurrentY = Y
End Sub

' 画像コピー
Function BmpCopy(sp, sx, sy, w, h, gp, gx, gy)
Dim exec
  exec = BitBlt(gp, gx, gy, w, h, sp, sx, sy, SRCCOPY)
End Function

' 画像重ね合わせコピー
Function BmpKasane(sp, mp, sx, sy, w, h, gp, gx, gy)
Dim exec
  exec = BitBlt(gp, gx, gy, w, h, mp, sx, sy, SRCPAINT)
  exec = BitBlt(gp, gx, gy, w, h, sp, sx, sy, SRCAND)
End Function

' 時間待ち
Function Delay(ByVal t As Long)
Dim At_0 As Long, At_1 As Long
  At_0 = GetTickCount
  At_1 = At_0
  While ((At_1 - At_0) < t)
    At_1 = GetTickCount
    DoEvents
  Wend
End Function

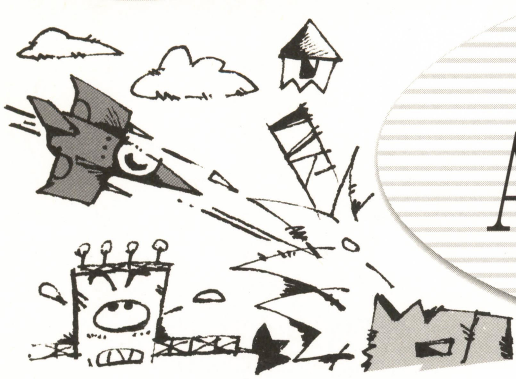
' 現在の絶対パス取得
Function MyForm()
  If Right$(App.Path, 1) <> "¥" Then
    MyForm = App.Path & "¥"
  Else
    MyForm = App.Path
  End If
End Function

' WAVE音声出力
Function PlayWAV(wavname As String, mode As Integer)
Dim sflag%, rc%
'
Select Case mode
    Case 0
        sflag% = 11 'SND_NODEFAULT + SND_ASYNC + SND_LOOP
    Case Is > 0
        sflag% = 3 'SND_NODEFAULT + SND_ASYNC
End Select
'演奏
    rc% = sndPlaySound(wavname, sflag%)
End Function
```

★Excelでプログラムが組めるなんて知りませんでした。Visual Basicもはじめたばかりですが，がんばって投稿できるレベルになろうと思います。（長崎市・**教師生活25年48歳**）……**【編：**学校の先生なんでしょうか。生徒にベーマガ読者がいるかもしれません。**】**

(超大作プログラム) リストが誌面に掲載できない作品です。
BMホームページ(http://www.basicmagazine.net/)からダウンロードしてお楽しみください。

Windows(Visual Basic V6.0+DirectX7)用



録画機能を持ったシューティング

アバンダンド

Abandoned Ver 1.01

hani（16歳））

はじめに

3ステージの横スクロールシューティング・ゲームです。

特徴としてはオプションが付き，それを切り離せること。敵の弾をアイテムに変換できる，バリアショットが撃てること。ゲームの録画が出できること。だと思います。

Aba.exeを実行したあと，Zキーでゲームをはじめます。Aba起動時には，起動している他のプログラムを，なるべくすべて終了させてください。

OpStart.exeをクリックして起動するといろいろ設定ができますので，それで設定することをオススメします。

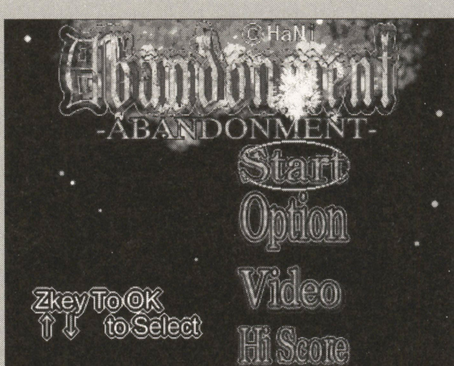
操作法

右手操作モード（初期設定）

↑キー…上移動
↓キー…下移動
←キー…左移動
→キー…右移動
Z キー…ショット，決定
X キー…バリアショット
C キー…オプション切り離し

左手操作モード

Wキー…上移動
A キー…左移動
D キー…右移動
S キー…下移動
←キー…ショット
↓キー…バリアショット
→キー…オプション切り離し



《写真１》高い完成度を誇るシューティング



《写真２》演出面や操作性，ゲーム内容もバッチリ



《写真３》プレイの録画機能も備えている



《写真４》ボス戦もある。バリアショットを発射だ！

ゲーム用語の説明

パワー…キャラのパワーアップアイム。
オプション…パワーをとると出現するキャラの周りにいる機体のミニチュア分身。
バリアショット…強力なショットとバリアを同時に放つ。
バリアバー…バリアショットを使える量を表示しているメーター。
ハート…画面下のキャラの残機を表すもの。
ショット…自機弾。

★先日，僕と12年間つきあってくれた「98RA」が壊れました。ひとえに影様のおかげです。諸君，まだ使えるパソコンを捨てちゃいかんよ。(群馬県高崎市・巽　洋平22歳）……【影：ずいぶん変わった名前の彼女だこと。しかも壊れてしまったのか？そりゃ大変だ。】

CHECKER FLAG

影：超大作ということもあるけど，完成度の高いシューティングだ。自キャラの回りにいるオプションの動きがうまくできているなぁ。また，ゲーム内容だけでなく，細かい設定ができるのも魅力の一つだ。

Dr.D：残念ながらリストが同封されていなかったので，リスト面での解説はできんが，録画機能を搭載しているのは，なかなかのものじゃ。こういう録画機能は，プレイヤーの操作を記録しておき，その記録した操作をソフト側で再現する，というのが基本じゃが，ゲームの形式によっては,うまく再現できなかったりするので，なかなか難しい部類だったりする。ぜひとも，この部分のリストは見たかったのぉ～。

編：超大作投稿でも，なるべくソースファイルの同封での投稿をお願いします。

ゲームのルール

敵の弾に当たるとハートが一つ減ります。ハートがなくなるとゲーム・オーバーです。

バリアショットはバリアバーがなくなるまで使えます。１ゲーム最高16回バリアショットを撃つことができます。バリアショットを撃つことにより，バリアバーからパワーを消費します。バリアショットのショットは，敵機にダメージを与えます。また，敵弾を消してパワーに変えます。

パワーをいくつかとると，オプションがつきます。オプションは何に当たっても壊れません。敵弾に当たり，ハートが残っていたらバリアバーは初期化されます。オプション切り離しは，かなり重要ですのでこまめに使うように。

録画機能について

ゲーム中にシフト・キーを押すとゲームの録画がスタートします。もう一度シフトキーを押すと録画データをセーブします。

録画データファイル名は何も指定しないと「UserFile」という名前になります。タイトルが表示されているときにVキーを押すと，自分で録画ファイルの名前の指定ができます。

録画したデータは，項目「Videos」で見ることができます。Videos項目を選択中，Xキーでデモプレイを，Cキーで通常録画ファイル（UserFile）を，Vキーで自分で名前をつけた録画ファイルのファイル名を入力することで，録画を再生できます。ファイル名などを何もいじらなければ，Cキーで録画が見れます。

（注意）録画中に敵の弾にあたると，そこまでの録画は無効になります。セーブされません。録画をはじめたが，気に入らない場合は弾にあたるか，プログラムを終了させてください。ボス戦，エンディングなどになったときは，強制的に録画をセーブします。セーブ，録画データ初期化時に時間がかかることがあります。

攻略

バリアショットを大胆に使うことが攻略の糸口です。

また，オプションなどすべてをうまく利用して，効率的にゲームをプレイしましょう。オプションもバリアショットについで非常に重要です。

バリアショットで消した敵弾は点数になります。これを考慮すればいい点数が取れると思います。

攻略は簡単で，撃たれる前に撃って，避けられる弾は避けて，避けられない弾はバリアショットで消すだけです。

Special Thanks!

刹那さんにアイデアなど非常に協力してもらいました。厨房（ハンドルネーム）がデザイン面で，ぶつくさ改良点を指摘してくれました。ありがとうございました！！

面倒くさかったので，一面の音楽はフリーのmidiを公開しているFinalia/P'sMATさんを利用しhttp://homepage1.nifty.com/pon-ta/finalia/

あと，２面の曲も適当なツールで適当に作りました。

最後に

え～，かなり適当なつくりになってしまいましたが，次回作は半端じゃなく力を入れています。いつになるかわからないですが，是非ページに遊びに来て，あったらダウンロードしてください。

URL:www.vb-game.ne.nu

メール:hani@katch.ne.jpです。

（お願い）男性声優を募集します。ドッターを募集します（RPG系。マップチップとか）。モンスターをかける人，お願い，描いてください。

背景描ける人を募集します。私１人だとやっぱダメです。限界です。助けてください。

ってゆ～か何でもいいので，ゲームに役立ちそうな技術を持ってたら，ぜひ力を貸してください。

★ようやくPHSの機種変更して，feel H"にしました。カラー画面にfeel Sound，も～う，最高です。影さんの携帯はどんな感じですか？（神奈川県川崎市・**桜井道孝21歳**）……【**影：**じつは私も編くんもfeel H"だったりする。ライトメール交換でもする？】

（超大作プログラム）　リストが誌面に掲載できない作品です。
BMホームページ(http://www.basicmagazine.net/)からダウンロードしてお楽しみください。

Windows(Visual Basic V6.0+DirectX7)用



3D対戦形式の戦車ゲーム

戦車（せんしゃ）

橘　一馬（29歳）

遊びかた

自分の戦車を操って敵の戦車をすべて破壊してください。

カーソル・キーで前進，後退，旋回ができます。コンコンってやると速くできます。スペース・キーで弾を発射します。

F1キーでカメラモードを切り替えます。何も押さないとレーダーが出ます。ジョイステックでも遊べます。

改造

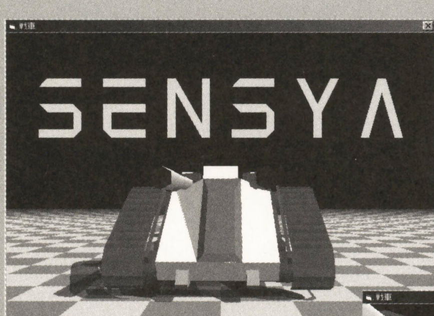
いろいろな性能の戦車を選べたら面白いかな…。壁とのあたり判定がうまくありません。どうすればいいのか，わかんないです。友達のノートパソコンでやったら，すんごく遅くてゲームになりませんでした。ウェイトをかけているので調整してみればいいかも…。また，フォッグをかけているので，はずしてみるとか，画面を小さくしてみるのもいいかも…。もっといい処理をすればよいのかな？VBの限界か？

★参考にしたもの

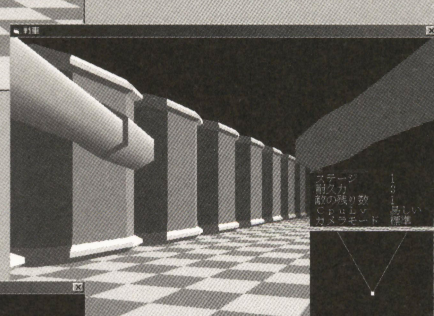
（効果音）～音と声のOnLine素材集～ WEBWAVELIB http://www10.freeweb.ne.jp/art /aissia /

（Xファイル）・プロジェクトチームDoG内DOGA-L　http://www.doga.co.jp/ptdoga/

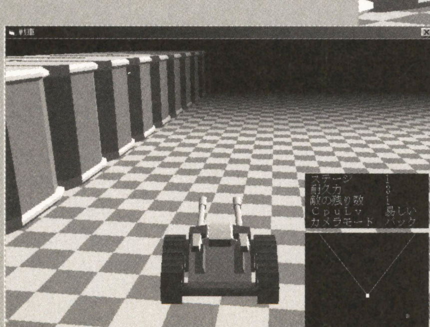
（プログラム）・Visual Basicによる DirectX7活用術　川田徹 著，・DirectX7 For Visual Basic　はじめるゲームプログラミング　藤田伸二 著



《写真１》3D戦車を操る対戦型のゲーム



《写真２》3Dならではの迫力ある視点



《写真３》敵の戦車を発見したら弾を発射しろ

CHECKER FLAG

つぐ美：完全3D形式の対戦型ゲームです。ベーマガでは，超大作も含めて数が少ない3Dプログラム・ゲームだけに，見た目が非常にいいわね。

影：ただ，ゲームとしては動作速度がキツくて，よほど速いマシンじゃないと遊べないなぁ～。説明文には「VBの限界か？」なんて書いてあるけど，言語に責任転嫁するようじゃダメ。こういう動作速度部分こそ，プログラマーの腕の見せどころだゾ。

編：昔の投稿プログラムは，言語というかハードの限界すらも超えていた作品がありましたよね。今では，ハードの性能がスゴすぎるため，そういうプログラムもなくなりましたが…。

Dr.D：VB+DirectXで，3Dゲームをしっかりと形にしているのは評価できるゾ。つぎは，ゲームとしての改良を期待したい。これは新しい作品を作るよりも，この作品を改造していくほうが能率がいいゾ。また，3Dプログラム自体の勉強もしていけば，当たり判定などについても，いろんな方法が見つかるはずじゃ。

★イラストが上手な人ってうらやましいですね。いや，まったく。（青森県南津軽郡・**ていこく22歳**）……【**影：**私のイラストをみんなが笑う。なぜ？**つぐ美：**影さんって，絵が下手という自覚がないんだから。あくせすROOMでは「うろおぼえ友の会」を実施中！】

Windows(HSP V2.5)用



走って，跳んで，逃げまくれ

走り跳び逃げ（はしとにげ）

INF（15歳）

内容

迫りくる敵からの妨害を避け，ひたすら逃げるゲームです。

丸い物体をかわせ!! 走って跳んで逃げろ!!

使いかたと遊びかた

HSPでリストを実行するか，ダウンロードファイルにある，hasiri.exeを起動してください。

操作方法

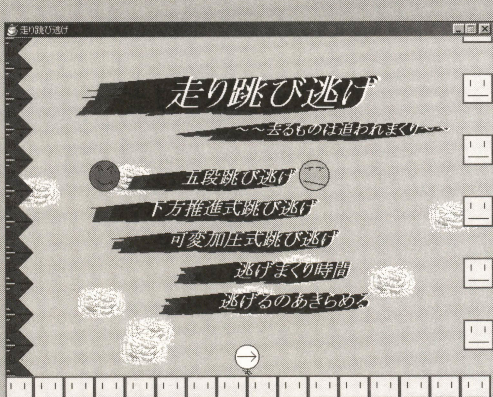
タイトル画面でカーソル・キー上下で選択，スペース・キーで決定。選択するものには，“五段跳び逃げ”，“下方推進式跳び逃げ”，“可変加圧式跳び逃げ”，“逃げまくり時間”，“逃げるのあきらめる”があります。

・**五段跳び逃げ**…その名のとおり5回連続でジャンプすることが可能なモードです。

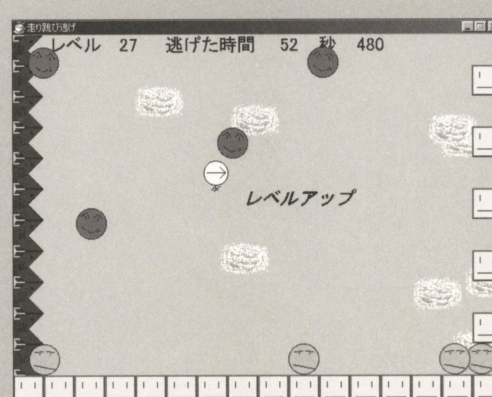
・**下方推進式跳び逃げ**…カーソル・キー上をおすと上昇，はなすと下降します。また，画面上に触れると，下に叩きつけられます。

・**可変加圧式跳び逃げ**…カーソル・キー上を押している長さによりジャンプする高さが変わります。

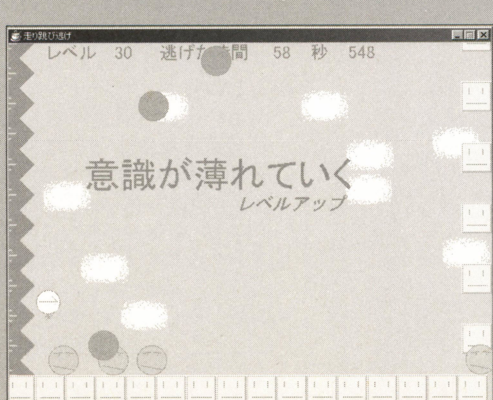
・**逃げまくり時間**…各逃げかたのうちで，



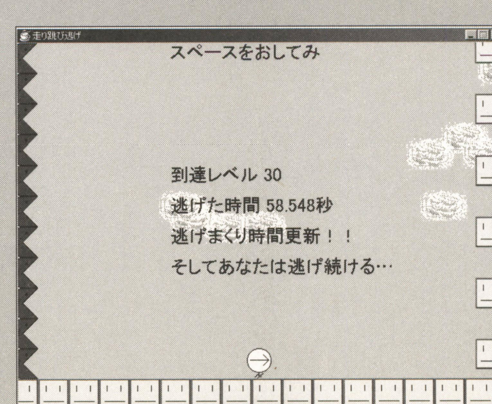
《写真1》操作の違いを持つ三つのモードを搭載



《写真2》敵や障害からひたすら逃げるゲーム



《写真3》画面外へ押し出されたりしたら終わり



《写真4》ハイスコアは，モードごとに保存される

★「プレステ2でゲームをプログラムする！」終わりなんですね。残念。（和歌山県橋本市・志賀 猛18歳）……【**つぐ美**：同じような内容のハガキが，他にもありました。連載の「BASIC STUDIO」入門のほうでがんばりますので，よろしくね。**やまかん**：よろしく！】

一番ながく逃げた時間を表示します。

・逃げるのあきらめる…逃げるのをあきらめてゲームを終了します。

ゲーム中は，カーソル・キー左右で移動，上でジャンプです。自キャラは，一定の速度で走っています。レベルが上がると，だんだん後ろに下がっていきます。

敵は自キャラの進行を妨害しようと前方に降ってきたり，落ちていたりしています（2種類だけですが)。地面に落ちている緑の敵にあたると，少しうしろにもどされ，はねている赤い敵にあたるとソコソコ後ろに戻されます。

2秒でレベルが1上がり，スピードが2追加されます。レベルが8上がるとスピードは初期値にもどり，敵の数が増えます。

どかどかと降ってくるブロックは，空中に道を作ってくれるので，一応味方です。ただし，あたるとゲーム・オーバーになってしまいます。

妨害を受けて後方の魚につかまると，ゲーム・オーバーです。最後に，どのくらい逃げたかなどが表示されます。

プログラムの仕組み

ハイスコアファイルがない場合は，自動生成されます。

プログラムの入力法と注意点

hasiri.bmpを用意してください。

中身は，自キャラ32×40を二つ，ブロック，魚，敵1，敵2をそれぞれ40×40で一つずつ，雲を64×40で一つ，それぞれ横に並べて背景を黒で保存してください。

wavファイルは，なんか適当に作って，それぞれ“jump.wav”,“hitfla.wav”“torya.wav”という名前にして保存してください。いらない場合は，リストの41，42，43行を注釈にするか消してください。

あとは，以上のファイルを同じフォルダに入れて，リストを入力すればOKす。

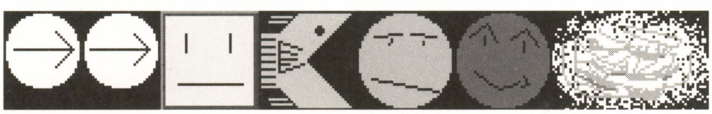
改造，その他

敵キャラを増やしたりとか，2人でバトルとかできるようにしてくれたらいいなぁ。あとベストなタイムを記録するところが一番めんどくさかったり。

小言

このプログラムは，とっとと作るつもりだったけれど，メンドクサイところがあったりして，なかなか時間がかかってしまいました。

本屋でベーマガを立ち読みしている らしい友人Y.M，読んでないで買え！



hasiri.bmpの画像

HSP

CHECKER FLAG

編：各モードとも基本的な内容は同じですが，キー入力の部分で違いがあるので，プレイするとチョットした感覚の違いが楽しめます。

つぐ美：レベルが上がるとスピードが初期化されるのは，賛否両論の意見がありそうですね。わたしとしては，スピードが上がり続けていく形のほうが面白いんじゃないかな，と思ったわ。

影：逆に，レベルが上がるごとにブレーキがかかることで，メリハリがついているとも言えるよね。

Dr.D：スクロールスピードについては，ゲーム性の問題なので，感覚の問題じゃな。読者自身で改造してみても面白いじゃろう。このゲームの特徴である，キー入力によるモード分けは，リストの面でも興味深い。キー入力はstick命令で行っており，ジャンプ部分でのif文の処理の違いで，特徴付けを行っておる。ただ，五段跳び式においては，stickのトリガー指定を変えており，押しっぱなしで反応しないように設定してある。

『走り跳び逃げ』のプログラム・リスト

```
;////////////////// 走り跳び逃げ ～～去るものは追われまくり～～ ///////////////
;--------------------------------------------------------- Programmed by INF

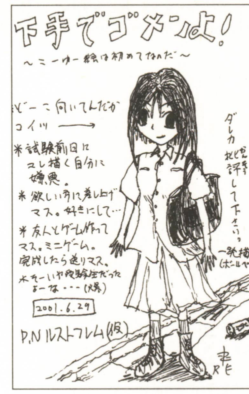
                                                 ;----------------- ファイルロード (highsc.hsc)
exist "highsc.hsc"
;////////// ファイルがなかったらファイル作成
if strsize=-1 {
        allhigher="001020000102000010200"
        bsave "highsc.hsc",allhigher
}
                                                 ;----------------- ファイルロードと数字に変換
bload "highsc.hsc",allhigher
strmid highfiverec,allhigher,0,7
strmid highgravrec,allhigher,7,7
strmid highplusrec,allhigher,14,7
                                                 ;------- 画像収納用の見えないスクリーンを用意
buffer 2,640,480,1       ;1を0に変えると少しきれいだけれど、フェードアウトができない
cls 4
picload "hasiri.bmp",1
                                                 ;----------------- タイトルの文字を作成
tit="走り跳び逃げ"
repeat 40
        color 0,0,255-(cnt*6)
        if cnt=39 : color 255,255,255
        font "ＭＳ Ｐ明朝",48,3
        pos cnt*3,50-(cnt/4) : mes tit
        font "ＭＳ Ｐ明朝",20,3
        pos 130+(cnt*2),110-(cnt/5) : mes "～～去るものは追われまくり～～"
        font "ＭＳ Ｐ明朝",24,3
        pos 61+(cnt*2),170-(cnt/5) : mes "五段跳び逃げ "
        pos 14+(cnt*2),210-(cnt/5) : mes "下方推進式跳び逃げ "
        pos 42+(cnt*2),250-(cnt/5) : mes "可変加圧式跳び逃げ "
        pos 130+(cnt*2),290-(cnt/5) : mes "逃げまくり時間 "
        pos 110+(cnt*2),330-(cnt/5) : mes "逃げるのあきらめる "
loop
                                                 ;----------------- 通常の画面を設定
screen 0,640,480,1       ;1を0に変えると少しきれいだけれど、フェードアウトができない
palcopy 2
title tit
                                                 ;----------------- 音をロード
sndload "jump.wav",0,0
sndload "hitfla.wav",1,0
sndload "torya.wav",2,0
gosub *cleara
```

★ポケコンで3Dポリゴンにチャレンジ！ポケコンでロボットを動かそう！ポケコン用ホームページ作成にチャレンジ！ポケコンでブロードバンドインターネットにチャレンジ！ポケコンで…むなしい。（島根県浜田市・**匿名工業高校生16歳**）……【**編：ははーっ。**】

```
*fititle        ;///// タイトル画面
        repeat
                gosub *mapdraw
                gosub *medraw
                                        ;------------- タイトルの文字とかをコピー
                pos 90,50 : gcopy 2,0,42,580,480
                font "ＭＳ　Ｐ明朝",24,3
                color 255,0,0
                pos 60+(mode*28)+50,mode*40+161
                gcopy 2,184,0,40,40
                pos 335+(mode*28)+50,mode*40+161
                gcopy 2,144,0,40,40
                stick key
                if (key=2)&(mode!0) : mode--
                if (key=8)&(mode!4) : mode++
                if key : snd 0
                if key=16 : break
                await waits
        loop
        ;/////// どれを選択したか
        if mode=3 : goto *highest
        if mode=4 : end
        repeat          ;-------------- さあ！走れ！！跳べ！！！そして逃げろ！！！！
                gosub *mapdraw
                gosub *timersetc
                gosub *itsme
                gosub *enehenyo
                gosub *enetorya
                gosub *medraw
                if death=1 : redraw 2 : snd 1 : break   ; ////「私はやられましたか？」
                await waits
                gosub *hitcheck
        loop                                    ;-------------------- ループ
                ;////////// やられると白くフェードアウト、そのあといろいろ処理
        font "ＭＳ　Ｐ明朝",48,0
        repeat 32
                palfade cnt*8
                await waits
                color 0,0,0 : pos 100,150 : mes "意識が薄れていく"
                redraw 1
        loop
        font "ＭＳ　Ｐ明朝",24,0
        mex=300 : mey=400
        gosub *scorewrite
        str timemilli,3
        repeat
                gosub *mapdraw
                gosub *medraw
                stick key
                if key&16 : event++
                palfade
                color 0,0,0
                if event>=1 : pos 200,160 : mes "到達レベル "+level
                if event>=2 : pos 200,200 : mes "逃げた時間 "+timesecond+"."+timemilli+"秒"
                pos 200,240
                if (event>=3) {
                        if (newrecord=1) : mes "逃げまくり時間更新！！" : else : mes "記録更新
ならず"
                }
                if event>=4 : pos 200,280 : mes "そしてあなたは逃げ続ける…"
                if event=5 : break
                pos 200,0 : mes "スペースをおしてみ"
        await waits
        loop
        gosub *cleara
        goto *fititle
*itsme                                  ;------------ 自キャラの処理
        mevx=6-speed
        if mode=0 : gosub *keystatefive
        if mode=1 : gosub *keystgrav
        if mode=2 : gosub *keystplus
        mex+=mevx
        if mex>608 : mex=608
        if mex<0 : mex=0
        mey+=jumpow
        if (mey<0)&(mode=1) : mey=0 : jumpow=60
        jumpow+=5
        if mey>400 : mey=400 : jumpow=1 : jpsp=0
        return
*timersetc                              ;-------------------- 時間関連およびレベルアップ
        if (timesecond¥2=0)&(enemy!30) {
                font "ＭＳ　ゴシック",24,3
                color 0,0,0 : pos 300,200
                mes "レベルアップ"
                if timemilli<waits {
                        level++
                        if (level¥8=1) : enemy++ : speed=2 : else : speed+=2
                }
        }
        timemilli+=waits
        if timemilli>=1000 : timesecond++ : timemilli-=1000
        font "ＭＳ　Ｐ明朝",24,0
        color 0,0,0
        pos 50,0 : mes "レベル" : pos 140,0 : mes level
        pos 200,0 : mes "逃げた時間"
        pos 350,0 : mes timesecond
        pos 400,0 : mes "秒"
        pos 450,0 : mes timemilli
        return
*keystatefive                                   ;---------- キーの処理(五段跳び式)
        stick key,5
        if key&1 : mevx-=12
        if key&4 : mevx+=12
        if (key&2) : if (jpsp!5) : snd 0 : jumpow=-26 : jpsp++
        return
*keystgrav                                      ;---------- キーの処理(下方推進式)
        stick key,7
        if key&1 : mevx-=12
        if key&4 : mevx+=12
        if (key&2) : jumpow-=8
        return
*keystplus                                      ;---------- キーの処理(可変加圧式)
        stick key,7
        if key&1 : mevx-=12
        if key&4 : mevx+=12
        if key&2 {
                if jpsp!10 : jumpow=-30 : jpsp++
                if jpsp=1 : snd 0
                } else : jpsp=10
        return
*mapdraw                                        ;---------- 雲と地面と追ってくるさかなを表示
        redraw 1 : redraw 2
        color 0,210,210 : boxf
        gmode 2
        repeat 9
                kumox.cnt-=cnt/3+speed
                if kumox.cnt<=-64 : kumox.cnt=640 : rnd kumoy.cnt,400
                pos kumox.cnt,kumoy.cnt : gcopy 2,226,0,64,40
        loop
        repeat 16
                pos rollx+(cnt*40),440 : gcopy 2,64,0,40,40
                pos rollx,cnt*40-200 : gcopy 2,104,0,40,40
                pos 600,(cnt*80)-(rollx*2)-840 : gcopy 2,64,0,40,40
        loop
        rollx-=speed
        if rollx<=-40 : rollx+=40
        return
*medraw                                         ;---------- 自キャラを表示
        if copx=0 : copx=32 : else : copx=0
        pos mex,mey : gcopy 2,copx,0,32,40
        return
*enehenyo                                       ;---------- 敵１（落ちてるやつ）
```

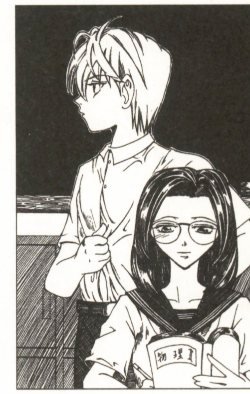

★ベーマガの9月号を買いそびれました。どうしたら手に入りますか？メールで注文してもいいですか？（奈良県生駒市・ホッチキス 15歳）……【編：webmas@basicmagazine.net　までメールで注文してください。または近くの本屋さんに注文してもOKです。】

評…うーん，なんともいえない味が出てると思うな（影）



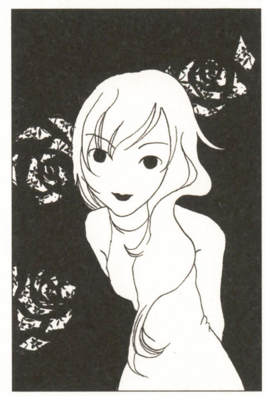
by ルストフレム

評…今回も制服キャラというわけで，よろしく！（影）



by 鈴木有人

評…見ようによってはちょっと怖いかもしれない（編）



by 比呂

評…「Catlikeのもっと身近に…」宛でしたが（編）



by 神無月

HSP

```
        repeat enemy
                if enhenx.cnt<-40 : rnd enhenx.cnt,100 : enhenx.cnt+=640
                enhenx.cnt-=speed
                pos enhenx.cnt,enheny : gcopy 2,144,0,40,40
        loop
        return
*enetorya                                       ;---------- 敵２（元気にはねまくっているやつ）
        repeat enemy
                if entorx.cnt<-40 : rnd entorx.cnt,580 : entorx.cnt+=340 : entory.cnt=-40 :
tojpow.cnt=0
                entory.cnt+=tojpow.cnt
                tojpow.cnt+=2 : entorx.cnt-=speed
                if entory.cnt>400 : entory.cnt=400 : tojpow.cnt=-39
                pos entorx.cnt,entory.cnt : gcopy 2,184,0,40,40
        loop
        return
*hitcheck                                       ;----------- あたり判定
        repeat enemy
                enefl=cnt
                repeat 4
                        htmx=mex-mevx+(mevx/4*(cnt+1))+5
                        htmy=mey-jumpow+(jumpow/4*(cnt+1))+5
                        repeat 2
                                if cnt=0 : htex=enhenx.enefl : htey=enheny
                                if cnt=1 : htex=entorx.enefl : htey=entory.enefl
                                if
((htmx+22)>=htex)&(htmx<=(htex+22))&((htmy+22)>=htey)&(htmy<=(htey+22)){
                                        mex-=6+cnt*6
                                        snd 2
                                }
                        loop
                loop
        loop
        if (mex>568)|(mex<(40-(rollx*-1))) : death=1
        return
*highest                                        ;------------ ベストな逃げ時間
        strmid frecs,highfiverec,0,4
        strmid frecm,highfiverec,4,3
        strmid grecs,highgravrec,0,4
        strmid grecm,highgravrec,4,3
        strmid precs,highplusrec,0,4
        strmid precm,highplusrec,4,3
        fmes=frecs+"."+frecm+"秒"
        gmes=grecs+"."+grecm+"秒"
        pmes=precs+"."+precm+"秒"
        repeat
                gosub *mapdraw
                gosub *medraw
                await waits
                color 0,0,0
                font "ＭＳ　Ｐ明朝",36
                pos 100,20 : mes "今までの最高逃げまくり時間"
                font "ＭＳ　Ｐ明朝",24
                pos 60,120 : mes "五段跳び式"
                pos 60,200 : mes fmes
                pos 260,120 : mes "下方推進式"
                pos 260,200 : mes gmes
                pos 460,120 : mes "可変加圧式"
                pos 460,200 : mes pmes
                pos 200,350 : mes "スペースを押すと戻る"
                stick key
                if key&16 : break
        loop
        goto *fititle
*cleara                                         ;----------- いろいろと変数を初期化、設定
        dim enhenx,30
        dim entorx,30 : dim entory,30 : dim tojpow,30
        dim kumox,9 : dim kumoy,9
        yazix=640 : yaziy=400
        death=0
        mex=300 : mey=400
```

```
        enemy=0
        jumpow=0
        event=0
        waits=41
        speed=2
        level=0
        timemilli=0 : timesecond=0
        randomize
        repeat 30
                rnd enhenx.cnt,100 : enhenx.cnt+=640 : enheny=400
                rnd entorx.cnt,640 : entory.cnt=-40
        loop
        repeat 9
                rnd kumox.cnt,640 : rnd kumoy.cnt,400
        loop
        return
*scorewrite                                     ;----------- 逃げた時間を書き込み（トップだけ）
        urecord=timesecond*1000+timemilli
        if mode=0 : erecord=highfiverec
        if mode=1 : erecord=highgravrec
        if mode=2 : erecord=highplusrec
        int erecord
        if (urecord>erecord) {
                newrecord=1
                if mode=0 : highfiverec=urecord
                if mode=1 : highgravrec=urecord
                if mode=2 : highplusrec=urecord
        }
        else : newrecord=0
        str highfiverec,7 : str highgravrec,7 : str highplusrec,7
        allhigher=highfiverec+highgravrec+highplusrec
        bsave "highsc.hsc",allhigher
        return
```

★う～，後期から連日9時間労働～！1回生ってみんな，こうなんですか？　それから，元気ですか，シャドウさん！（大阪府河内長野市・**よっち18歳**）……【**編：何事も最初が肝心！がんばれ！大学入りたくても入れない人だっているんだから，弱音を吐かずに！**】

Windows(F-BASIC V6.0)用



四字熟語を知らないと遊べないゾ

四字熟語（よじじゅくご）

伊藤　優（18歳）

内容

画面に表示される漢字を並べ替えて，正しい四字熟語を完成させるゲームです。全問正解すると所要時間を表示して終了します。

制限時間がなくなると，コンティニュー選択になります。コンティニューした場合は，制限時間が10秒増えます。

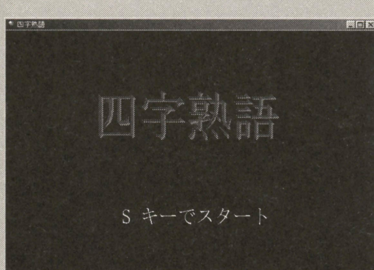
遊びかた・操作法

1キー…1文字目と2文字目を入れ替えます。

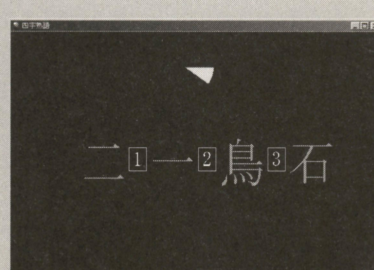
2キー…2文字目と3文字目を入れ替えます。

3キー…3文字目と4文字目を入れ替えます。

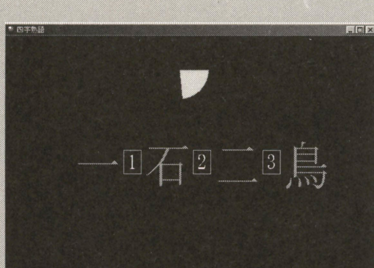
スペース・キー…並べ替えてから押すと，正解かどうか判定を行います。



《写真1》タイトルどおり，四字熟語を扱うゲーム



《写真2》これくらいは，楽勝だよね？



《写真3》これが正解。影さんみたいなボケはしないように！



《写真4》制限時間内に多くの四字熟語を完成させよう

CHECKER FLAG

Dr.D：文字の入れ替えが，主な処理となるためswap命令を使えば，簡単に実現できるゲームじゃな。その単純な構成が，リストの短さにつながっておる。ただ，問題の文字コード化は，答えを見られないための処理じゃろうが，問題の追加がやりにくいという弊害を生んでしまったな。

つぐ美：知識がないと，遊べないゲームですよね。影さんみたいな人は全然楽しめないゲームね。

影：何を言うんだ。こんなゲーム，オレ様なら1プレイで全クリだぜ。さっそくプレイだ。Sキーを押してスタートっと，第1問目は…カチッ…カチッ……カチッ……アレッつぎへ進まない？。

編：影さん，「一二鳥石」なんて四字熟語はありませんよ。

正解の場合は，つぎの問題に移ります。

プログラムについて

問題を追加するときは，JIS命令文で漢字のコードを調べて，DATA部分に追加してください。つぎにMONDAI=20となっている部分を，問題数と一致するように書き換えてください。

改造については，英単語などに拡張すると面白いかもしれません。

★参考にしたもの
- 過去のベーマガ
- FINAL FANTASY Ⅶ（スクウェア）

★10月号は「編集部の学校訪問記」がありませんでしたが，夏休みだったから？うちのコンピュータ部は夏休みの3分の1は活動してましたよ。（三重県津市・げいじゅつ15歳）……【編：学校訪問記の取材希望校を募集中です。メールや電話でお申し込みください。】

『四字熟語』のプログラム・リスト

```
randomize(time) '乱数系列初期化

dim QUE(20,4),ANS(20,4)
common shared FLAG

declare sub KEYDRAW(MON,ANS())
declare sub HANTEI(MON,QUE(),ANS())

MONDAI=20:MON=1:FLAG=1:LIM=60000  '問題数,制限時間

symbol(153,103),"四字熟語",5,5,1  'タイトル
symbol(150,100),"四字熟語",5,5,3
symbol(200,300),"S キーでスタート",2,2,15
while A$<>"s":A$=inkey$:wend
for M=1 to MONDAI:for X=1 to 4  '問題データのロード
  read QUE(M,X):ANS(M,X)=QUE(M,X)
next:next

*START
A&=millitime

while millitime-A&<LIM and MON<=MONDAI  'メインループ
 while FLAG=1
   for X=1 to 10:swap ANS(MON,rnd*3+1),ANS(MON,rnd*3+1):next 'シャッフル
   HANTEI MON,QUE(),ANS()
 wend
 cls:KEYDRAW MON,ANS() '画面描画

while FLAG=0 and millitime-A&<LIM
 A$=inkey$
 CHA=-1*(A$="1")-2*(A$="2")-3*(A$="3")
 if CHA>0 then
   swap ANS(MON,CHA),ANS(MON,CHA+1) '入れ替え
   cls:KEYDRAW MON,ANS()
 end if
 if A$=" " then HANTEI MON,QUE(),ANS()
 A=(millitime-A&)/LIM+0.01
 if A>1 then A=1
 circle(300,60),50,5,,0,A,f,pset,3 '時間表示
wend
 if MON=MONDAI and FLAG=1 then CLEAR=1
 MON=MON+1:FLAG=1
while A$<>"":A$=inkey$:wend 'キーバッファクリア
wend

 if CLEAR=0 then
   cls:symbol(15,125),"時間切れ CONTINUE? (Y/N)",3,3,13
   while A$="":A$=inkey$:wend
   if A$="y" then LIM=LIM+10000:mon=1:goto *START  '再スタート
 end if
 if CLEAR=1 then
   B&=millitime
   cls:symbol(125,125),"ゲームクリア",4,4,3
   symbol(75,205),"所要時間"+str$((B&-A&)/1000)+"秒",4,4,3
end if
while A$<>"":A$=inkey$:wend 'キーバッファクリア
stop:end

'問題データ
data 12396,16464,18035,17467
data 15732,14676,19509,15983
data 18255,15402,17772,18807
data 12396,13604,21098,16430
data 19253,15459,19509,16205
data 17772,19291,16446,16758
data 16469,18212,17726,12871
data 14925,16167,14195,18551
data 15155,15207,16229,19520
data 18476,19069,18558,16205
data 20021,17964,15448,18552
data 14206,17754,15685,19816
data 12904,20021,17728,25165
data 18772,13926,16983,17719
data 15424,18807,16215,19819
data 19520,13888,15199,16229
data 18484,15155,13112,16420
data 13115,16949,12396,16168
data 16440,15444,18732,19527
data 16185,19813,19324,15965

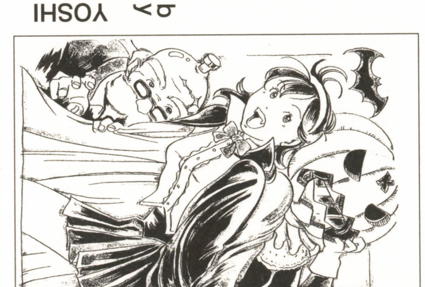
sub HANTEI(mon,QUE(),ANS()) '判定
FLAG=1
for X=1 to 4
FLAG=FLAG*(ANS(MON,X)=QUE(MON,X))
next
end sub

sub KEYDRAW(MON,ANS()) '画面描画
for X=1 to 4:symbol(120*X,180),knj$(ANS(MON,X)),5,5,5:next
for X=1 to 3
 symbol(120*X+75,200),str$(X),2,2,15
 line(120*X+84,199)-(120*X+114,239),pset,15,b
next
end sub
```



by 村田春佳

評…早く続きが見たいんですけど…よろしくね（つぐ美）

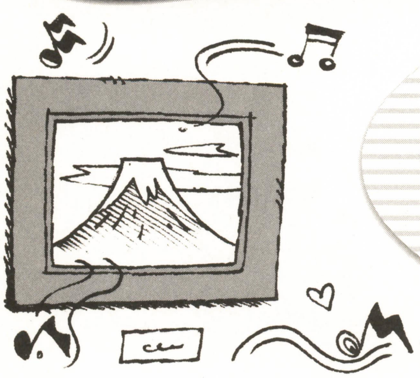


by YOSHI

評…もうすぐハロウインですね。時期ネタGOOD（つぐ美）

★秋は運動会。我が家もマウスが大騒ぎ！（愛知県名古屋市・**最馬場可？歳**）……【**影**：運動会の時期は，パソコンの活躍の場も増えるし，家のネズミも暴れまわるという，すばらしい句だ。**編集長**：我が編集部のネズミ男はいつも暴れまわってるが…。**影**：私？】

Windows(QBasic)用



テキストファイルから音楽再生

Translation Composer 翻訳作曲家

（トランスレーション コンポーザー ほんやくさっきょくか）

石山　明弘（33歳）

内容

最近，学校教育の総合学習において，創意工夫の重要性が指摘されています。そこで，生徒１人１人の描いた絵画を翻訳して音楽を作り出す「Translation Composer」プログラムを作成しました。

絵画はアルファベットの“O”で描き，縦12行×横40文字のテキストファイルで描きます。

このシステムは，その絵画ファイルを読み込み，翻訳処理して，まずまずの音楽を作り出します。この作り出した音楽を編曲したりして，創意工夫の授業の教材にしてください。

このシステムは，テキストファイルですが，絵画のもつ感受性から，新たに音楽が作り出されるので，生徒の感受性の育成や，創意工夫の手助けをする教材としては，大変良いと思います。

「車」や「富士山」や「家」の絵画から試しに演奏してみましたが，まずまずの音楽が作り出され，まるで魔法のようです。

リストには，「ノーマル」や「メジャー」，「マイナー」，「沖縄」バージョンがあり，聴いていてなかなか面白いです。

音のステップ数にもよりますが，12×40の絵画で，最大２の480乗とおりのランダム音楽を作曲しますので，このシステムの想像力は，かなり無限大に近いのではないでしょうか？

Version.1では，まだまだですが，将来，画像エディタや，Composerシステムフォームを開発すれば，クリエイターの手助けになるかもしれません。

操作方法

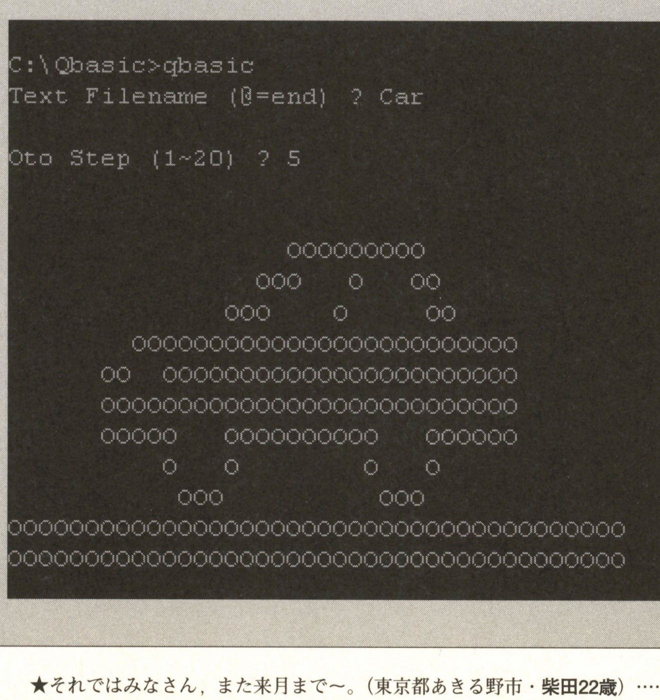
MS-DOSからQBasicを起動します。**リスト１**のプログラムを入力もしくはロードして実行すると，システムはつぎのように聞いてきます。

"Text Filename (@=end) ";

ここで，絵画ファイル名を入力します。"@"で，システム終了です。

つぎに，"Oto Step (1~20) ";と聞いてきます。ここで，半音上がる“O”文字数を入力します。

するとシステムは，絵画ファイルから翻訳を開始し，読み込んだ絵画を念のため表示し，最後に翻訳した音楽を演奏します。演奏後，絵画ファイル入力待ちになります。



《写真１》テキストの違いで，再生される音楽も異なる

苦労や改造のアドバイス

作曲処理の部分の文字列処理や，オクターブ音階データの試行錯誤に苦労しました。

絵画テキストの横幅や縦幅を替えたり，本格的な画像エディタを開発するとよいでしょう。

半音上がるステップ数や，１オクターブの音階データを色々と替えてお試しください。

画像エディタを含めて，半音上がるステップ数や，１オクターブの音階データを色々と替えられる本格的なComposerシステムフォームをVBなどで開発するとよいでしょう。

面白い音楽ができたり，より良いオク

★それではみなさん，また来月まで〜。（東京都あきる野市・柴田22歳）……【**影**：はい，今月も多くのお便りありがとうございました。来月は11月8日発売です。お楽しみに！**つぐ美**：ちょっと，まだページあるんですけど。これ以降，影さん登場させないわよ。】

《第１表》変数表

HX = 40	絵画テキストの横幅
HY = 12	絵画テキストの縦幅
Mus$(HY)	作曲した音楽データ
ST	Oto $ の音階を半音(+1)上がるステップ数（ST= 1 /絵画テキストの数）
OP = 0 + ST	現在の音程ポインタ（値= 0 ～ 1 2）
Oto$(12)	1オクターブの音階データ(C,C#…としないでもよい）
Namae $	絵画ファイル名
(GX,GY)	絵画の解釈位置
Jp $	1 オクターブの上下フラグ
Moji $	絵画の 1 文字読み込み
Bun$,Bun2 $	絵画の再生成変数
Mu $	1 行毎の音楽生成変数

《第２表》音階データについて

沖縄バージョンの音階データ例

Oto $ (0) = "c"	Oto $ (1) = "e"	Oto $ (2) = "f"
Oto $ (3) = "g"	Oto $ (4) = "b"	Oto $ (5) = "g"
Oto $ (6) = "f"	Oto $ (7) = "g"	Oto $ (8) = "f"
Oto $ (9) = "e"	Oto $ (10) = "f"	Oto $ (11) = "g"

1 音階=「ド・ミ・ファ・ソ・シ・ソ・ファ・ソ・ファ・ミ・ファ・ソ」

その他のバージョンは 3 半音毎に，コード「C，F，G，C」や，コード「Am，Dm，Em，Am」，コード「C，F，G7，C」とすると，なぜか良いようです。

作者は，エレクトーン 6 級までは取ったのですが，作曲の基本は，コード「C，F，G7，C」の組み合わせだということを教わりました。多分そうだと思いますが，かなり昔に習ったので，大分忘れました(笑)。

ターブデータが発見できたら，下記のアドレスにメールをもらえるとうれしいです。クリエイターの仕事になるかもしれないし，相乗効果が期待できますので。

CHECKER FLAG

つぐ美：投稿の中には，沖縄バージョンやマイナーバージョンといった，音のデータを変更したリストもあったのですが，都合によりノーマルバージョンのみリストを掲載しています。他のバージョンは，ＨＰ上のデータに収録してあります。また，絵画ファイルにも，いろんなパターンがありましたが，こちらもcar.txtのみの掲載で，他のファイルはＨＰ上のデータ収録です。

編：表の中で，沖縄バージョンの音階データは掲載しているので，沖縄バージョンへの変更は，誌面の情報からでも可能ですね。

Dr.D：発想はなかなかのものじゃが，やはりテキストで絵画ファイルを作成するのが，ちと面倒じゃな。また，QBasicのBeep音では，よほどの偶然がない限り，しっかりとした音楽になりにくい印象じゃな。Windowsで作成すれば，同じアイデアで，MIDIから再生が可能になるので，より音の面での広がりが期待できるゾ。

影：市販されている作曲ソフトでも，同じような機能を搭載しているソフトをみたことがあるなぁ～。

《リスト１》『Translation Composer　翻訳作曲家』のプログラム・リスト

```
REM*****************************************************
REM***Translation Composer ver.1************************
REM*****Copyright (C) 2001.7.26 by Ishiyama Production***
REM**URL : http://www24.freeweb.ne.jp/computer/i480pro/**
REM**E-mail : i480pro@mx9.freecom.ne.jp*****************
REM*************************************Normal Version***
HX = 40: '---Text Width----
HY = 12: '---text Height---
REM---Set Onkai------------------------------------------
DIM Mus$(HY):  '---Music---
ST = 1 / 10: '---Hanon Step---
OP = 0 + ST: '---Oto Pointer---
DIM Oto$(12): '---Onkai---
Oto$(0) = "c": Oto$(1) = "e": Oto$(2) = "g": '---Code C
Oto$(3) = "c": Oto$(4) = "f": Oto$(5) = "a": '---Code F
Oto$(6) = "d": Oto$(7) = "g": Oto$(8) = "b": '---Code G
Oto$(9) = "c": Oto$(10) = "e": Oto$(11) = "g": '-Code C
REM***LOOP**********************************************
ROOP:
REM---Read Draw Text & Compose Music---------------------
PRINT "Text Filename (@=end) ";
INPUT Namae$: IF Namae$ = "@" THEN END
Namae$ = Namae$ + ".txt": PRINT
Roop2:  PRINT "Oto Step (1~20) ";
        INPUT ST: IF ST < 1 THEN GOTO Roop2
ST = ABS(ST) MOD 21: ST = 1 / ST
OP = 0 + ST: PRINT
'------------------------------
OPEN Namae$ FOR INPUT AS #1
FOR GY = 1 TO HY
        LINE INPUT #1, Bun$: Bun2$ = "": Mu$ = ""
        FOR GX = 1 TO HX
                Moji$ = MID$(Bun$, GX, 1)
                IF Moji$ = "" OR Moji$ = " " THEN
                                Bun2$ = Bun2$ + " "
                                OP = OP - ST: Jp$ = ""
                                IF OP < 0 THEN Jp$ = "<"
                                IF OP < 0 THEN OP = OP + 12
                                Mu$ = Mu$ + Jp$ + Oto$(INT(OP))
                        ELSE
                                Bun2$ = Bun2$ + "0"
                                OP = OP + ST: Jp$ = ""
                                IF OP >= 12 THEN Jp$ = ">"
                                IF OP >= 12 THEN OP = OP - 12
                                Mu$ = Mu$ + Jp$ + Oto$(INT(OP))
                        END IF
        NEXT GX
        PRINT Bun2$: Mus$(GY) = Mu$
NEXT GY
CLOSE #1
REM---Play Music-----------------------------------------
PLAY "l48o5"
FOR GY = 1 TO HY
        PLAY Mus$(GY)
NEXT GY
'-----GOTO LOOP-------------
GOTO ROOP
REM*****************************************************
REM***End of System*************************************
REM*****************************************************
```

《第３表》簡単なフローチャート

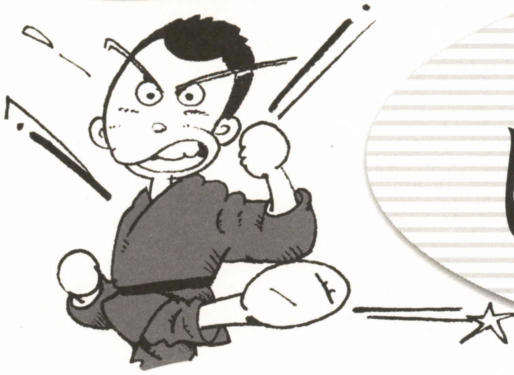
音階のセット
↓
LOOP
↓
絵画テキストの読み込み＆作曲
↓
演奏
↓
GOTO LOOP（LOOPの場所へ戻る）

《リスト２》絵画ファイル例（car.txt）

```
         000000000
       000    0    00
      000     0      00
    0000000000000000000000000
  00  000000000000000000000000
    0000000000000000000000000
    00000    0000000000    000000
       0    0          0    0
        000               000
0000000000000000000000000000000000000000
0000000000000000000000000000000000000000
```


★ハガキとインターネットハガキでは，どちらが掲載される確率が高いですか？（山梨県韮崎市・**新沼19歳**）……【**影：**それはですね…。**つぐ美：**だから，登場しないでください。**影：**そんなー。**つぐ美：**どちらの掲載率が高いということはありません。よろしく！】

iアプリ用



懐かしい形式の対戦格闘

ぽんこつファイト

Mr．親父（18歳）

作るきっかけ

ある日，ネットの波をさまよっていると，懐かしいゲームが目に飛び込んできた。そのゲームの名は「ジョイメカファイト」!!　天下の任天堂が，ファミコン時代に出した格闘ゲームである。よし，これをiアプリで再現してみよう！！

CHECKER FLAG

編：iアプリによる対戦格闘です。昔懐かしの雰囲気をもつ画面やキャラクターの動きが，個人的には好きです。
つぐ美：ただ，ゲーム内容は，必殺技を連射すれば勝てるという形になってしまっているため，面白味には欠けますね。
Dr.D：変数が多いリストのため，変数表がほしかったところじゃな。また，チトif文が多いリストになっているのが気になるのぉ。switch文や，キー部分でいえば，iKeyの値をうまく使ってif文を省略すれば，もっとスッキリしたリストになるじゃろう。

操作方法

→：自キャラが右に移動
←：自キャラが左に移動
↓：自キャラがしゃがむ（意味なし）
2：弱パンチ
3：弱キック
5：強パンチ
6：強キック
必殺技（青弾（飛び道具））：↓→2

CPUについて

CPUの思考ルーチンは，かなり卑屈なキャラ設定になっています。こんな思考ルーチンです。

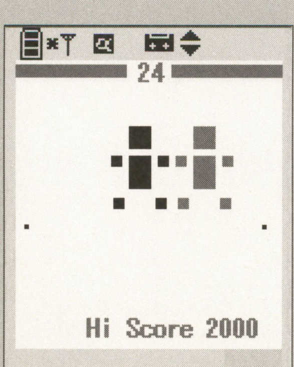
1：自キャラが2秒動かなければ，近づいてきて弱パンチ後，逃げる。
2：自キャラが近づいてくると逃げる。
3：追い詰められると弱パンチを連打する。

いろいろ

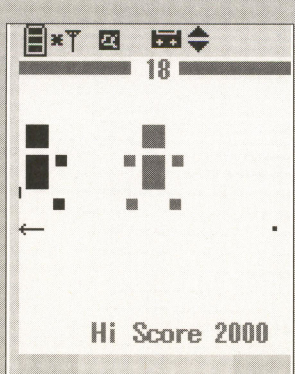
今回もi-jadeでデバッグしただけで実機では動かしてません。早く自分専用のi-modeが欲しいです。

今回，このプログラムを作るにあたって，当たり判定にはベーマガ2001年1月号の「Catlikeのもっと身近にプログラミング！」のコーナーで紹介されていた方法を使いました。また，コマンド入力の方法としてベーマガ2001年6月号の「GET&RUN」のコーナーで紹介されていた方法を使いました。しかし，「思い出のゲームを再現する」と言いながらまったく別の物になってしまいました。でも，iアプリって良いですね。比較的簡単にこんな物が作れちゃうんだから。

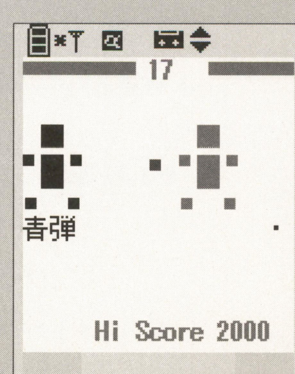
余談ですが「ｉａｐｐｌｉを「イアピー」って呼ぼうリング」という物を作ってみました。興味のある人は下記のURLまで来てください。http://www10.freeweb.ne.jp/play/mr_oyaji/iapy.htm



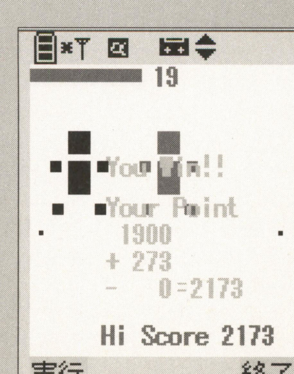
《写真1》iアプリで対戦格闘を実現



《写真2》単純な画像ながらも，しっかり動く



《写真3》必殺技で勝負が決まるゾ



《写真4》KO勝ちを目指してプレイ

★後輩たちがうらやましいと思う今日この頃…。教訓，受験勉強はお早めに。（徳島市・氷月りおん17歳）……【つぐ美：ふだんからしっかり勉強していれば，受験勉強であわてることもないと思うけどなー。影：私は，切羽詰らないと力を発揮できないタイプです。】

《リスト１》fight_l.javaのプログラム・リスト

```
import com.nttdocomo.util.*;
import com.nttdocomo.ui.*;
import com.nttdocomo.io.*;
import java.io.*;
import java.util.*;
import java.lang.*;
import javax.microedition.io.*;

public class fight_l extends IApplication
{
        public void start()
        {
                FightScreen fs = new FightScreen();
                Display.setCurrent(fs);
        }
}

class FightScreen extends Canvas implements Runnable{
        int key_flg = 0,key_flg2 = 0;
        int e_key_flg = 0,e_key_flg2 = 0,e_key_flg3 = 0;
        int waza_flg1 = 0;
        int width = 0,height = 0;
        int second = 0,def_second = -1;
        int head_x = 0,head_y = 0,head_w = 0,head_h = 0;
        int r_leg_x = 0,r_leg_y = 0,hi_k = 0;
        int r_hand_x = 0,hi_p = 0;
        int e_head_x = 0,e_head_y = 0,e_head_w = 0,e_head_h = 0;
        int e_r_leg_x = 0,e_r_leg_y = 0;
        int e_r_hand_x = 0;
        int e_move = 0;
        int waza1 = 0,waza1_x = 0;
        int power = 0,e_power = 0;
        int point = 0,e_point = 0;
        int p_win = 0,e_win = 0;
        int time = 30;
        int temp_e_head_x = 0,e_time = 0;
        int start_key = 0,end_key = 0;
        int score = 0,hiscore = 0;
        String key = "",e_key = "";
        String time_space = "",plus_space = "",minus_space = "",hiscore_space = "";
        Thread myThread;

        public FightScreen(){
                width = getWidth();
                height = getHeight();
                head_x = 30;e_head_x = 80;
                head_y = 30;e_head_y = 30;
                head_w = 10;e_head_w = 10;
                head_h = 10;e_head_h = 10;
                setSoftLabel(Frame.SOFT_KEY_2,"開始");
                try{
                    DataInputStream in = Connector.openDataInputStream("scratchpad:///0");
                    hiscore = in.readInt();
                    in.close();
                }catch(IOException e){}
                if (hiscore <= 0)
                        hiscore = 2000;

                myThread = new Thread(this);
                myThread.start();
        }

        public void run(){
                int iKey = 0;

                for(;;){
                    iKey = getKeypadState();
                    if (iKey==(1<<Display.KEY_SOFT2) && (time>0 && p_win==0 && e_win==0))
                        {       // 開始
                                start_key = 1;
                                setSoftLabel(Frame.SOFT_KEY_2,"");
                        }
                    else if(iKey==(1< Display.KEY_SOFT1) && (time<=0 || p_win==1 || e_win==1))
                        {// 再実行      // 変数の初期化
                                key_flg = 0;key_flg2 = 0;
                                e_key_flg = 0;e_key_flg2 = 0;e_key_flg3 = 0;
                                waza_flg1 = 0;
                                width = 0;height = 0;
                                second = 0;def_second = -1;
                                head_x = 0;head_y = 0;head_w = 0;head_h = 0;
                                r_leg_x = 0;r_leg_y = 0;hi_k = 0;
                                r_hand_x = 0;hi_p = 0;
                                e_head_x = 0;e_head_y = 0;e_head_w = 0;e_head_h = 0;
                                e_r_leg_x = 0;e_r_leg_y = 0;
                                e_r_hand_x = 0;
                                e_move = 0;
                                waza1 = 0;waza1_x = 0;
                                power = 0;e_power = 0;
                                point = 0;e_point = 0;
                                p_win = 0;e_win = 0;
                                time = 30;
                                temp_e_head_x = 0;e_time = 0;
                                start_key = 0;end_key = 0;
                                score = 0;hiscore = 0;
                                key = "";e_key = "";
                                time_space = "";plus_space = "";
                                minus_space = "";hiscore_space = "";
                                width = getWidth();
                                height = getHeight();
                                head_x = 30;e_head_x = 80;
                                head_y = 30;e_head_y = 30;
                                head_w = 10;e_head_w = 10;
                                head_h = 10;e_head_h = 10;
                                setSoftLabel(Frame.SOFT_KEY_1,"");
                                setSoftLabel(Frame.SOFT_KEY_2,"開始");
                                try{
DataInputStream in = Connector.openDataInputStream("scratchpad:///0");
                                        hiscore = in.readInt();
                                        in.close();
                                }catch(IOException e){}
                                if (hiscore <= 0)
                                        hiscore = 2000;
                                repaint();
                        }
                    else if (iKey==(1<<Display.KEY_SOFT2) && (time<=0 || p_win==1 || e_win==1))
                                // 終了
                                IApplication.getCurrentApp().terminate();

                        if (start_key == 1)
                        {
                                if (iKey==(1 << Display.KEY_UP)) // 上
                                        key_flg = 1;
                                else if (iKey==(1 << Display.KEY_DOWN))
                                {       // 下
                                        key_flg = 2;
                                        if (waza_flg1 == 0)
                                                waza_flg1 = 1;
                                }
                                else if (iKey==(1 << Display.KEY_LEFT))
                                        // 左
                                        key_flg = 3;
                                else if (iKey==(1 << Display.KEY_RIGHT))
                                {       // 右
                                        key_flg = 4;
                                        if (waza_flg1 == 1)
                                                waza_flg1 = 2;
                                }
                                else if (iKey==(1 << Display.KEY_2))
                                {       // 弱突
```

i アプリ

★MSXで組んだプログラムでも，投稿受け付けていますか？夏休みはずっと，MSXをいじくっていました。プログラムもちょっと組んでみました。(青森県八戸市・**C.C.C14歳**)……**【編：とてもすばらしい作品であればMSXだって掲載可能です。参考にもなりますし。】**

```
					key_flg = 5;
					if (waza_flg1 == 2)
						waza_flg1 = 3;
				}
				else if (iKey==(1 << Display.KEY_3)) // 弱蹴
					key_flg = 6;
				else if (iKey==(1 << Display.KEY_5)) // 強突
					key_flg = 7;
				else if (iKey==(1 << Display.KEY_6)) // 強蹴
					key_flg = 8;
				else
				{
					key_flg = 0;
					Calendar cal = Calendar.getInstance();
					second = cal.get(Calendar.SECOND);
					if (second != def_second || def_second == -1)
					{
					if ((time>0 && p_win!=1 && e_win!=1) && def_second!=-1)
							time--;
						if (def_second != -1)
							e_time++;
						def_second = second;
						waza_flg1 = 0;
						if (e_time > 10)
							e_time = 2;
					}
				}

				if (key_flg != 0)
					e_time = 0;
				repaint();
			}
		}
	}

	public void paint(Graphics g) {
		int d_head_y = 0;
		int d_body_y = 0;
		int l_hand_x = 0,d_hand_y = 0;
		int l_leg_x = 0,l_leg_y = 0;
		int e_d_head_y = 0;
		int e_d_body_y = 0;
		int e_l_hand_x = 0,e_d_hand_y = 0;
		int e_l_leg_x = 0,e_l_leg_y = 0;

		// 動き // CPU
		e_key_flg = 0;
		e_r_leg_y = 0;
		e_key = "  ・";
		if (e_key_flg3 == 1)
		{
			if (e_head_x < (head_x + 40) && (e_head_x + 13) < (width))
			{
				e_key = "  →";
				e_head_x++;
			}
			else
			{
				e_key_flg3 = 0;
				e_time = 0;
			}
			if ((e_head_x % 4) == 1)
				e_l_leg_y = -5;
			else if ((e_head_x % 4) == 3)
				e_r_leg_y = -5;
		}
		else if (e_time >= 2)
		{
			if ((head_x + head_w) < (e_head_x - 10))
			{
				e_key = "  ←";
				e_head_x--;
			}
			if ((e_head_x % 4) == 1)
				e_l_leg_y = -5;
			else if ((e_head_x % 4) == 3)
				e_r_leg_y = -5;
		}
		e_key_flg2++;
		if ((head_x + head_w) >= (e_head_x - 10) && e_key_flg2 > 1)
		{	// 弱突
			e_key = "弱突";
			e_key_flg = 5;
			e_l_hand_x = -5;
			e_key_flg2 = 0;
			e_key_flg3 = 1;
		}
		if (temp_e_head_x > 0)
		{
			e_head_x = temp_e_head_x;
			temp_e_head_x = 0;
		}
	else if ((key_flg==5 || key_flg==6 || key_flg==7 || key_flg==8)&& key_flg2 != 1)
		{
			e_key = "  →";
			temp_e_head_x = e_head_x;
			if (key_flg == 5)
				e_head_x++;

			else if (key_flg == 6)
				e_head_x++;

			else if (key_flg == 7)
				e_head_x += 2;

			else if (key_flg == 8)
				e_head_x += 2;

			if ((e_head_x % 4) == 1)
				e_l_leg_y = -5;

			else if ((e_head_x % 4) == 3)
				e_r_leg_y = -5;

		}

		// PLAYER
		if (hi_k != 1 && hi_k != 2)
			r_leg_y = 0;

		if (key_flg == 1)
			key = "↑";
		else if (key_flg == 2)
		{
			key = "↓";
			d_head_y = 3;
			d_body_y = 3;
			d_hand_y = 3;
		}
		else if (key_flg == 3)
		{
			key = "←";
			if ((head_x - 3) > 0)
				head_x--;

			if ((head_x % 4) == 1)
				l_leg_y = -5;
			else if ((head_x % 4) == 3)
				r_leg_y = -5;
```


★つぐ美さんとはちがい，老人のまま永久に年をとらないDr.Dは哀れだ。（千葉県柏市・鈴木有人28歳）……【**Dr.D：**だれが哀れじゃ。ワシはこれからが花じゃ。**編：**ところで，Dr.Dって何歳なんですか？**Dr.D：**そんなことはワシも忘れた。それでいいのじゃ。】

```
}
else if (key_flg == 4)
{
        key = "→";
        if ((head_x + 13 + 5) < e_head_x)
                head_x++;

        if ((head_x % 4) == 1)
                l_leg_y = -5;

        else if ((head_x % 4) == 3)
                r_leg_y = -5;
}
else if (key_flg == 5 && hi_p == 0 && hi_k == 0)
{
        if (key_flg2 != 1)
        {
                key = "弱突";
                l_hand_x = 5;
                key_flg2 = 1;
        }
        else
                key = "・";
}
else if (key_flg == 6 && hi_p == 0 && hi_k == 0)
{
        if (key_flg2 != 1)
        {
                key = "弱蹴";
                l_leg_x = 5;
                l_leg_y = -5;
                key_flg2 = 1;
        }
        else
                key = "・";
}
else if ((key_flg == 7 || hi_p != 0) && hi_k == 0)
{
        if (key_flg2 != 1)
        {
                key = "強突";
                if (hi_p == 0)
                        hi_p = 1;
                else if (r_hand_x < 30 && hi_p == 1)
                {
                        head_x++;
                        r_hand_x += 5;
                        if (r_hand_x >= 30)
                                hi_p = 2;
                }
                else
                {
                        r_hand_x += -5;
                        if (r_hand_x < 0)
                        {
                                key_flg2 = 1;
                                r_hand_x = 0;
                                hi_p = 0;
                        }
                        else
                                head_x--;
                }
        }
        else
                key = "・";
}
else if ((key_flg == 8 || hi_k != 0) && hi_p == 0)
{
        if (key_flg2 != 1)
        {
```

```
                key = "強蹴";
                if (hi_k == 0)
                        hi_k = 1;
                else if ((30 + r_leg_y) > 0 && hi_k == 1)
                {
                        head_x++;
                        r_leg_x += 5;
                        r_leg_y += -5;
                        if ((30 + r_leg_y) <= 0)
                                hi_k = 2;
                }
                else
                {
                        r_leg_x += -5;
                        r_leg_y += 5;
                        if (r_leg_y > 0)
                        {
                                key_flg2 = 1;
                                r_leg_x = 0;
                                r_leg_y = 0;
                                hi_k = 0;
                        }
                        else
                                head_x--;
                }
        }
        else
                key = "・";
}
else
{
        key = "・";
        key_flg2 = 0;
}

if (waza_flg1 == 3)
{
        key = "青弾";
        waza1 = 1;
}

g.lock();
g.clearRect(0,0,width,height);

// キャラ描画　// CPU
g.setColor(g.getColorOfRGB(255,0,0));
g.fillRect(e_head_x ,e_head_y+e_d_head_y ,e_head_w ,e_head_h);                              // 頭
g.fillRect(e_head_x ,e_head_y+13+e_d_body_y ,e_head_w ,e_head_h+5);                         // 体
g.fillRect(e_head_x-8+e_l_hand_x ,e_head_y+13+e_d_hand_y ,e_head_w-5,e_head_h-5);           // 左手
g.fillRect(e_head_x+13+e_r_hand_x ,e_head_y+13+e_d_hand_y,e_head_w-5,e_head_h-5);           // 右手
g.fillRect(e_head_x-7+e_l_leg_x ,e_head_y+32+e_l_leg_y ,e_head_w-5,e_head_h-5);             // 左足
g.fillRect(e_head_x+12+e_r_leg_x ,e_head_y+32+e_r_leg_y ,e_head_w-5,e_head_h-5);            // 右足

// PLAYER
g.setColor(g.getColorOfRGB(0,0,0));
g.fillRect(head_x ,head_y + d_head_y ,head_w ,head_h);                                       // 頭
g.fillRect(head_x ,head_y + 13 + d_body_y ,head_w ,head_h + 5);                              // 体
g.fillRect(head_x - 8 + r_hand_x ,head_y + 13 + d_hand_y ,head_w - 5,head_h - 5);            // 右手
g.fillRect(head_x + 13 + l_hand_x ,head_y + 13 + d_hand_y,head_w - 5,head_h - 5);            // 左手
g.fillRect(head_x - 7 + r_leg_x ,head_y + 32 + r_leg_y ,head_w - 5,head_h - 5);              // 右足
g.fillRect(head_x + 12 + l_leg_x ,head_y + 32 + l_leg_y ,head_w - 5,head_h - 5);             // 左足

        if (waza1 == 1 && (head_x + 13 + waza1_x) < width + 5)
        {       // 青弾
                g.setColor(g.getColorOfRGB(0,0,255));
                g.fillRect(head_x + 13 + waza1_x ,head_y + 15 ,5 ,5);
                waza1_x++;
        }
        else if ((head_x + 13 + waza1_x) > width + 5)
        {
```


★最近デジカメを買った。毎日何かしら撮影しまくっているが，どれも消すのは惜しくて，僕のパソコンがJPEG画像だらけになっていく！（東京都三鷹市・カメラ小僧3号16歳）……**【つぐ美：その気持ちわかるわ。項目別などに整理することも楽しいものよ。】**

```
			wazal = 0;
			wazal_x = 0;
			waza_flgl = 0;
	}
	g.setColor(g.getColorOfRGB(0,0,0));
	g.drawString(key,0,80);
	g.drawString(e_key,95,80);

	// 当たり判定　// CPU
	if (e_key_flg == 5)
	{
		int e_top = 0, e_bottom = 0, e_left = 0, e_right = 0;
		int p_top = 0, p_bottom = 0, p_left = 0, p_right = 0;
		int top = 0, bottom = 0, left = 0, right = 0;
		int over = 0;
		if (e_key_flg == 5)
		{
			// 弱突(左手)
			e_top = e_head_y + 13 + e_d_hand_y;
			e_bottom = e_head_y + 13 + e_d_hand_y + e_head_h - 5;
			e_left = e_head_x - 8 + e_l_hand_x;
			e_right = e_head_x - 8 + e_l_hand_x + e_head_w - 5;
		}
		p_top = head_y + d_head_y;
		p_bottom = head_y + 13 + d_body_y + head_h + 5;
		p_left = head_x;
		p_right = head_x + head_w;
		if (e_top > p_top)
			top = e_top;
		else
			top = p_top;
		if (e_bottom < p_bottom)
			bottom = e_bottom;
		else
			bottom = p_bottom;
		if (e_left > p_left)
			left = e_left;
		else
			left = p_left;
		if (e_right < p_right)
			right = e_right;
		else
			right = p_right;
		if (top <= bottom && left <= right)
			over = 1;
		else
			over = 0;

		if (over == 1)
		{
			if (e_key_flg == 5) // 弱突
				e_power++;
		}
	}

	// PLAYER
	if (key_flg==5 || key_flg==6 || key_flg==7 || key_flg==8 || wazal==1)
	{
		int p_top = 0, p_bottom = 0, p_left = 0, p_right = 0;
		int e_top = 0, e_bottom = 0, e_left = 0, e_right = 0;
		int top = 0, bottom = 0, left = 0, right = 0;
		int over = 0;
		if (key_flg == 5)
		{	// 弱突(左手)
			p_top = head_y + 13 + d_hand_y;
			p_bottom = head_y + 13 + d_hand_y + head_h - 5;
			p_left = head_x + 13 + l_hand_x;
			p_right = head_x + 13 + l_hand_x + head_w - 5;
		}
		else if (key_flg == 6)
```

```
		{	// 弱蹴(左足)
			p_top = head_y + 32 + l_leg_y;
			p_bottom = head_y + 32 + l_leg_y + head_h - 5;
			p_left = head_x + 12 + l_leg_x;
			p_right = head_x + 12 + l_leg_x + head_w - 5;
		}
		else if (key_flg == 7)
		{	// 強突(右手)
			p_top = head_y + 13 + d_hand_y;
			p_bottom = head_y + 13 + d_hand_y + head_h - 5;
			p_left = head_x - 8 + r_hand_x;
			p_right = head_x - 8 + r_hand_x + head_w - 5;
		}
		else if (key_flg == 8)
		{	// 強蹴(右足)
			p_top = head_y + 32 + r_leg_y;
			p_bottom = head_y + 32 + r_leg_y + head_h - 5;
			p_left = head_x - 7 + r_leg_x;
			p_right = head_x - 7 + r_leg_x + head_w - 5;
		}

		if (wazal == 1)
		{	// 青弾
			p_top = head_y + 15;
			p_bottom = head_y + 15 + 5;
			p_left = head_x + 13 + wazal_x;
			p_right = head_x + 13 + wazal_x + 5;
		}
		e_top = e_head_y + e_d_head_y;
		e_bottom = e_head_y + 13 + e_d_body_y + e_head_h + 5;
		e_left = e_head_x;
		e_right = e_head_x + e_head_w;
		if (p_top > e_top)
			top = p_top;
		else
			top = e_top;
		if (p_bottom < e_bottom)
			bottom = p_bottom;
		else
			bottom = e_bottom;
		if (p_left > e_left)
			left = p_left;
		else
			left = e_left;
		if (p_right < e_right)
			right = p_right;
		else
			right = e_right;
		if (top <= bottom && left <= right)
			over = 1;
		else
			over = 0;

		if (over == 1)
		{
			if (key_flg == 5) // 弱突
				power++;
			else if(key_flg == 6) // 弱蹴
				power++;
			else if(key_flg == 7) // 強突
				power += 2;
			else if(key_flg == 8) // 強蹴
				power += 2;
			if (wazal == 1)
			{	// 青弾
				power += 30;
				wazal = 0;
				wazal_x = 0;
				waza_flgl = 0;
			}
```


★昔々，五反田に，影と編のへんな兄弟がいたとな。その兄弟は，病気がちなDr.Dおじいちゃんのために，毎日汗を流して働いていた。そこへ，娘・つぐ美がやってきて…。あとはおまかせします。(東京都江東区・**オッス15歳**)……**【編**：というわけで，読者におまかせ。**】**

```
		}
}
// メータ		// CPU
g.setColor(g.getColorOfRGB(255,255,0));
g.fillRect(width - 50 + (power / 5) ,0 ,50 ,10);
g.setColor(g.getColorOfRGB(255,0,0));
g.fillRect(width - 50 + (power / 5) ,2 ,50 ,6);
// PLAYER
g.setColor(g.getColorOfRGB(255,255,0));
g.fillRect(0 ,0 ,50 - (e_power / 10) ,10);
g.setColor(g.getColorOfRGB(255,0,0));
g.fillRect(0, 2 ,50 - (e_power / 10) ,6);

g.setColor(g.getColorOfRGB(255,0,0));
if (time < 10)
{
	g.drawString("0" + Integer.toString(time),55,10);
	g.drawString("0" + Integer.toString(time),56,10);
}
else
{
	g.drawString(Integer.toString(time),55,10);
	g.drawString(Integer.toString(time),56,10);
}

g.setColor(g.getColorOfRGB(255,175,0));
if (time <= 0)
{
		g.drawString("Time Over",34,50);
		g.drawString("Time Over",35,50);
}
else
{
	if ((width - 50 + (power / 5)) >= width && (e_win == 0))
	{
		g.drawString("You Win!!",34,50);
		g.drawString("You Win!!",35,50);
		p_win = 1;
		// 得点
		g.drawString("Your Point",34,68);
		g.drawString("Your Point",35,68);
		if (end_key == 0)
		{
			point = power;
			e_point = e_power;
		}
	}
	else if ((50 - (e_power / 10)) < 0 && (p_win == 0))
	{
		g.drawString("You Lost...",30,50);
		g.drawString("You Lost...",31,50);
		e_win = 1;
	}

	if (time >= 0 && time < 10)
		time_space = " ";
	if (point >= 0 && point < 10)
		plus_space = "   ";
	else if (point >= 10 && point < 100)
		plus_space = "  ";
	else if (point >= 100 && point < 1000)
		plus_space = " ";
	if (e_point >= 0 && e_point < 10)
		minus_space = "   ";
	else if (e_point >= 10 && e_point < 100)
		minus_space = "  ";
	else if (e_point >= 100 && e_point < 1000)
		minus_space = " ";

if (p_win == 1)
	{
		g.drawString(time_space + Integer.toString(time * 100),40,80);
		g.drawString(time_space + Integer.toString(time * 100),41,80);
		g.drawString("+" + plus_space + Integer.toString(point),34,92);
		g.drawString("+" + plus_space + Integer.toString(point),35,92);
		g.drawString("-" + minus_space + Integer.toString(e_point),34,104);
		g.drawString("-" + minus_space + Integer.toString(e_point),35,104);
		g.drawString("=" + Integer.toString(time*100+point-e_point),66,104);
		g.drawString(" " + Integer.toString(time*100+point-e_point),67,104);
		score = time * 100 + point - e_point;
	}
	}

	if ((time <= 0 || p_win == 1 || e_win == 1) && end_key == 0)
	{
		setSoftLabel(Frame.SOFT_KEY_1,"実行");
		setSoftLabel(Frame.SOFT_KEY_2,"終了");
		end_key = 1;
	}

	if (score > hiscore)
	{
	  try{
	    DataOutputStream out=Connector.openDataOutputStream("scratchpad:///0");
	    out.writeInt(score);
	    out.close();
	    hiscore = score;
	  }catch(IOException e){}
	}

	if (hiscore >= 0 && hiscore < 10)
		hiscore_space = "   ";
	else if (hiscore >= 10 && hiscore < 100)
		hiscore_space = "  ";
	else if (hiscore >= 100 && hiscore < 1000)
		hiscore_space = " ";
    g.setColor(g.getColorOfRGB(255,0,0));
    g.drawString("Hi Score "+hiscore_space+Integer.toString(hiscore),width-88,height-5);
    g.drawString("Hi Score "+hiscore_space+Integer.toString(hiscore),width-87,height-5);
	g.unlock(false);
    }
}
```

《リスト２》fight_l.jamのリスト

```
AppName = fight_l
PackageURL = fight_l.jar
AppSize = 4875
AppClass = fight_l
LastModified = Fri, 22 Jun 2001 00:06:26
SPsize = 20
```

《リスト３》HTMLファイルの記述例

```
<html>
<HEAD>
<META http-equiv="Content-Type" content="text/html; charset=Shift_JIS">
<TITLE>fight_l</TITLE>
</HEAD>
<body>


<OBJECT declare id="fight_l" data="fight_l.jam" type="application/x-jam">
</OBJECT>
<A ijam="#fight_l" href="index.htm">ﾀﾞｳﾝﾛｰﾄﾞ</A>

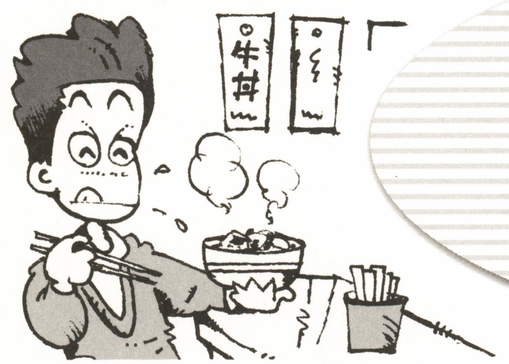

</body>
</html>
```

★家の電化製品はずっと昔から松下製です。そこに，兄貴の部屋にSONYのベガ（テレビ）が入りました。みんな，兄貴の部屋でテレビ見てます。（岩手県一関市・**松下16歳**）……【**編**：ということは，ゲーム機も「3DO」ってこと？そこに「PS2」がやってきて…。】

iアプリ用

牛食えで落ち物パズル

牛丼工場（ぎゅうどんこうじょう）



Hase（16歳）

内容

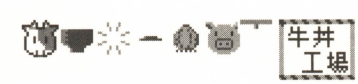
『牛丼食えスト』向けに制作していたミニゲームを，規模縮小して作ったｉアプリです。

ゲームの内容は，牛丼の具を丼ではさんで消す，というものです。

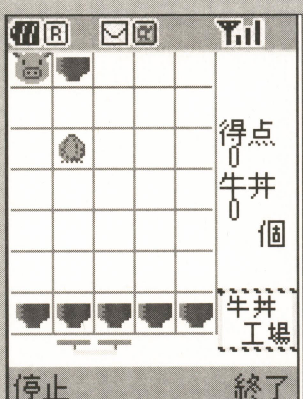
入力方法

画像ファイルを用意→テキストエディタでリストを入力→コンパイル→Preverifyで検証→Jamで圧縮→エミュレータ上で実行（もしくは，サイトにアップロードし，携帯にダウンロード）という手順になります。詳しくは，ベーマガに連載中の「君にもできるｉアプリ制作」を見てください。

画像ファイルは，左からCow.gifDon.dif,Erased.gif，Futa.gif,Negi.gif,Pig.gif（それぞれ16×1ドット），Machine.gif（16×2ドット），Title.gif（33×26ドット）となっています（**画像ファイル図**参照）。



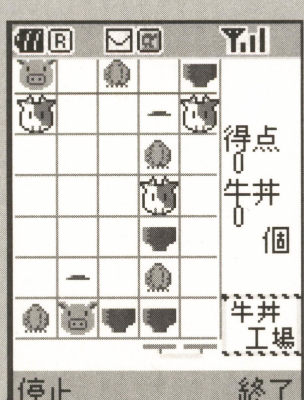
画像ファイル図



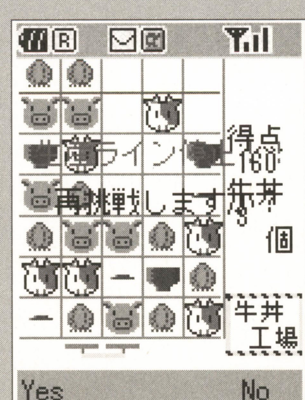
《写真１》牛丼を題材にしたパズル



《写真２》タマネギ，牛を組み合わせる



《写真３》それを丼で挟めば，牛丼の一丁上がり



《写真４》上まで積み上げてしまうとダメ

実行方法

503iシリーズ（携帯電話）・エミュレータのどちらでも動くはずです。

方法１…ｉアプリ端末エミュレータに読み込んで実行します。代表的なものにｉ－JADE，JAVA2ME Wireless SDK for DoJaなどがあります。

方法２…JAVA対応のｉモード携帯電話にダウンロードする方法です。jar，jamファイルと，jamへのリンクを含むHTMLをサイトにアップロードしてください。そうしたら，携帯電話にURLを入力し，リンクを選択すれば，ダウンロードされるはずです。

遊びかた

プログラムを実行するとすぐ，ゲームが始まります。作業ライン（材料の山）最下段の自機を左右キーで移動させ，決定キーで列を入れ替えて，上から落ちてくる材料を積み上げてください。

牛（牛肉だと思ってください）とタマ

★5インチ・ドライブ接続！読まない…。（福島市・南斗ライバ25歳）……【影：うーん，読めない，読めない。Dr.D：どうかしたか。ファイルが壊れてるのか？影：いや，この文字が読めなくて困ってるんです。Dr.D：アラビア語？ぎゃふん。影：？？？】

ネギを丼とフタではさむことで，牛丼を作ります。はさむ具の数が多い，つまり大盛りにすると，高得点になります。ひとつでも豚を入れると減点です。

さらに，同じ材料を縦に二つ重ねると消えるようになっています。得点にはなりませんが，じゃまなもの（豚・下に丼のない具など）をどけるのに便利です。赤い線まで材料を積み上げてしまうとゲーム・オーバーです。

ルールは大体こんなものです。作った牛丼の数がある程度増えると，材料の落ちる速度が上がり，牛丼を作ったときの得点も大きくなります。高得点を目指してみてください。

プログラムのしくみ

基本的には，「君にもできるｉアプリ制作」のサンプルと同じような構造です。GFCanvasクラスのスレッドを生成し，Runメソッドの無限ループ内で処理を行っています。

とくに高度なことや複雑なことはしていません（スレッドぐらい？）。強いてひとつ，変わったことを挙げれば，～？～：～という演算子ぐらいでしょうか。これは，たとえば“A = (B== 3)? 1：2；”というように書き，この例なら，Bが3なら1，それ以外なら2がAに代入されます。ちょっとした計算で，if文を使わずに条件ごとに値を変えることができるので，結構便利だと思います。ちなみに，JAVAだけでなくCやC++でも使えるようです。

《第1表》フィールド・メソッド表
（GFCanvasクラス）

メソッド	
GFCanvas()	コンストラクタ
run()	スレッドのメイン処理
subInit()	ゲーム開始時の初期化
paint(Graphics g)	画面の描画
getImg(String sName)	画像の読み込み
フィールド(変数)	
img～	キャラクター画像
iDon	牛丼製造数
iKey, iKeyPrev	キー状態・1ループ前のキー状態
iMx	自機の位置
iSeq	移動カウント
iSpd	落下速度
iScore	得点
strMap	仮想マップ(D:丼 F:フタC:牛 N:タマネギ P:豚)
bPause	ポーズフラグ
bOver	ゲームオーバーフラグ
TIPW, TIPH	画像サイズ
MAPW, MAPH	仮想マップサイズ

苦労した点

一言で言えば，「とにかく小さく作る」ことです。10Kバイトの容量制限はいうまでもなく，表示するキャラクターのサイズ，最低限ゲームに必要なものの選択などです。

その結果，キャラクターのサイズは16ドット，タイトル画面・ハイスコア表示などはなしという現在の形になりました。

コメント

ｉアプリは初挑戦になります。Javaに関しては，以前にちょっとしたものを組んだことがあるだけで，まともに使うのはこれが初めてです。

ｉアプリの容量制限などもあり，幾分わかりにくいプログラムになったかと思いますが，何かの参考になれば幸いです。

もともとこのゲームは，DirectXで組む予定でしたが，「規模を小さくすればｉアプリになるな」と思い，計画を変更しました。実際に作ってみて，これぐらいのゲームであれば，ｉアプリは手ごろな環境だと思いました。「ゲームは作りたいけど難しそうだ」と思っている方は，一度挑戦してみてはいかがでしょうか。

★参考にしたもの

・ベーマガ4～7月号『君にもできるｉアプリ制作』

・月刊JAVAワールド（IDG Japan刊）7月号『使えるiアプリの作り方』

CHECKER FLAG

編：まず，このリストの修正について。送られてきたリストでは，実機では動作しなかったため，リスト内に修正部分を記述しておきました。修正部分は,swap()メソッドにある，replace命令です。これは修正しないと，実機だけでなくJDK1.3+DoJaの環境でもエラーになってしまいます。実機でテストできない人は，最低でもNTT公認のDoJa環境でのテストをお願いします。

Dr.D：ゲームとしては，しっかりと形になってはいるが，内容は某落ち物パズルそっくりなため独自性に欠ける。また，キー入力部分が連続して判定されるため，遊びにくい。これについては，10月号のベストプログラム『牛丼道』のキー入力部分が，参考になるはずじゃ。ぜひ目を通してみてほしい。

《リスト1》GFApp.javaのプログラム・リスト

```
import com.nttdocomo.ui.*;
import com.nttdocomo.util.*;

/*
 * タイトル:  GyudonFactory
 * 説明:    牛食えiアプリ:牛丼工場
 * 作者: Hase
 * version 1.0
 */

public class GFApp extends IApplication {
  GFCanvas canvas;
  public GFApp() { // コンストラクタ
    canvas = new GFCanvas();
  }
  public void start() {
    Display.setCurrent(canvas);
  }
}
```

《リスト2》GFCanvas.javaのプログラム・リスト

```
import java.util.*;
import com.nttdocomo.ui.*;
import com.nttdocomo.io.*;

/*
 * タイトル:  GyudonFactory
 * 説明:    牛食えiアプリ:牛丼工場
 * 作者 : Hase
 * version 1.0
 */

public class GFCanvas extends Canvas implements Runnable{
  final int TIPW = 16, TIPH = 16; // 画像サイズ
  final int MAPW = 5, MAPH = 7; // マップ広さ(画面にあわせて調整可)

  Thread myThread;
  Image imgDon, imgFuta, imgPig, imgCow, imgNegi; // 画像 (材料)
```


★おもしろかった記事やつまらなかった記事などを，数字で選ぶようにしたのがいい。クターがいい。（福島市・**羽曽部　健15歳**）…
…【**影：**「クターのアスレチックワールド」にはまってます。なかでも，「クターの大根」最高！グハハハ。なぜ3秒台がでるんだ！】

★新ゲーム機Xboxはどう略すのだろう。まさか，「えくぼ」？ビル・ゲイツはここまで考えてなかったのかなぁ。(兵庫県神戸市・JANDY18歳）……【編：編集部でも話題になったことがあるんだよ。「ペケボ」「ぺけばこ」「クスボ」…。いつのまにか話題変わっちゃったけど。】

```
Image imgTitle, imgMachine, imgErased; // 画像 (自機・タイトルなど)

int iSeq, iSpd, iKey, iKeyPrev; // 状態・速度・キー情報
int iScore, iDon, iDonPrev, iMx; // 得点・製造数・レベル・自機の位置

boolean bPause, bOver; // ポーズフラグ, ゲームオーバーフラグ
StringBuffer strMap = new StringBuffer(); // 仮想マップ
Random rnd = new Random(); // 乱数

public GFCanvas() { // コンストラクタ
  imgMachine = getImg("resource:///Machine.gif");
  imgTitle = getImg("resource:///Title.gif");
  imgDon = getImg("resource:///Don.gif");
  imgFuta = getImg("resource:///Futa.gif");
  imgPig = getImg("resource:///Pig.gif");
  imgCow = getImg("resource:///Cow.gif");
  imgNegi = getImg("resource:///Negi.gif");
  imgErased = getImg("resource:///Erased.gif");

  subInit();

  // 乱数初期化
  Calendar cal = Calendar.getInstance();
  rnd.setSeed(cal.get(12));

  myThread = new Thread(this);
  myThread.start();
}

public Image getImg(String sName) { // 画像読み込みメソッド(『チープ・インパクト』より)
  MediaImage mi = MediaManager.getImage(sName);
  try {
    mi.use();
  } catch (ConnectionException e) {
  }
  return mi.getImage();
}

public int strPos(int iX, int iY) { // 文字列位置取得(『チープ・インパクト』より)
  return iX + iY * MAPW;
}

public void subInit() { // ゲームスタート
  int iCnt;
  strMap.delete(0, strMap.length());
  for (iCnt = 0; iCnt < MAPW * (MAPH - 1); iCnt++)
    strMap.append(' '); // マップクリア
  for (iCnt = 0; iCnt < MAPW; iCnt++)
    strMap.append('D'); // 最下段

  setSoftLabel(Frame.SOFT_KEY_1, "停止");
  setSoftLabel(Frame.SOFT_KEY_2, "終了");

  bPause = bOver = false;
  iMx = 1;
  iSpd = 15;
  iSeq = iSpd / 2;
  iScore = iDon = 0;
  setNext();
}

public void pause(boolean bOnOff) { // ポーズ・解除
  bPause = bOnOff;
  setSoftLabel(Frame.SOFT_KEY_1, bPause? "再開" : "停止");
}

void swap() { // 列の入れ替え
  int i;
  StringBuffer strBuf = new StringBuffer();

  if ((iKey == (1 << Display.KEY_SELECT)) && (iKeyPrev != (1 << Display.KEY_SELECT))) {
```

```
      //修正前
      //strBuf.replace(0, strBuf.length(), strMap.toString());   ↓
      //修正後
      strBuf.append( strMap.toString()) ;
    for (i = MAPH - 1; i > 0 && (strBuf.charAt(strPos(iMx    , i)) != ' ' ||
                                 strBuf.charAt(strPos(iMx + 1, i)) != ' '); i--) {
      strMap.setCharAt(strPos(iMx    , i), strBuf.charAt(strPos(iMx + 1, i)));
      strMap.setCharAt(strPos(iMx + 1, i), strBuf.charAt(strPos(iMx    , i)));
    }
  }
}

void setNext() {  // 次の材料
  int i;
  String strItem = new String("  C  N  D  F  C  N  P"); // ここを書き換えることで材料の出現率
を調整可能
  for (i = 0; i < MAPW; i++)
    strMap.setCharAt(strPos(i, 0), strItem.charAt(Math.abs(rnd.nextInt() %
strItem.length())));
}

void startFall() {
  int i;
  for (i = 0; i < MAPW; i++) {
    if (strMap.charAt(strPos(i, 1)) == ' ') {
      strMap.setCharAt(strPos(i, 1), strMap.charAt(strPos(i, 0)));
      strMap.setCharAt(strPos(i, 0), ' ');
    }
  }
}

void hantei() {
  int i, j, k;
  int icow = 0, inegi = 0, ipig = 0;
  boolean bfuta = false, bdon = false; // 判定用
  // 消去判定 (隣接)
  for (i = 0; i < MAPH - 1; i++) {
    for (j = 0; j < MAPW; j++) {
      if (strMap.charAt(strPos(j, i)) != ' ' && strMap.charAt(strPos(j, i)) != 'E') {
        if (strMap.charAt(strPos(j, i)) == strMap.charAt(strPos(j, i + 1))) {
          strMap.setCharAt(strPos(j, i + 1), 'E');
          strMap.setCharAt(strPos(j, i    ), 'E');
        }
      }
    }
  }

  // 消去判定 (牛丼)
  for (i = 0; i < MAPW; i++) {
    for (j = MAPH - 1; j > 0; j--) {
      if (strMap.charAt(strPos(i, j)) == 'D') {
        bdon = true;
        icow = inegi = ipig = 0;
        bfuta = false;
        for (k = j - 1; k >= 0 && bdon == true && bfuta == false; k--) {
          switch (strMap.charAt(strPos(i, k))) {
          case 'C':
            icow++;
            break;
          case 'N':
            inegi++;
            break;
          case 'P':
            ipig++;
            break;
          case 'F':
            bfuta = true;
            break;
          default:
            bdon = false;
            break;
```

```
        }
      }
      if (bfuta == true && bdon == true) {
        // 消える
        for (k+=1; k < j + 1; k++)
          strMap.setCharAt(strPos(i, k), 'E');
        // 得点計算 (大盛りだとボーナス、豚丼は減点)
        iScore += (ipig > 0)? -50 : ((icow > 0) && (inegi > 0)? (25 - iSpd) * (icow +
inegi) * 2: 0);
        if (iScore < 0)
          iScore = 0;
        iDon += (ipig > 0)? 0 : ((icow > 0) && (inegi > 0)? 1 : 0); // 生産個数
      }
    }
  }
}

int fall() { // 落下
  int i, j, k;
  k = 0;
  for (i = MAPH - 2; i > 0; i--) {
    for (j = 0; j < MAPW; j++) {
      if (strMap.charAt(strPos(j, i)) != ' ' && strMap.charAt(strPos(j, i + 1)) == ' ') {
        strMap.setCharAt(strPos(j, i + 1), strMap.charAt(strPos(j, i)));
        strMap.setCharAt(strPos(j, i    ), ' ');
        k++;
      }
    }
  }
  return k;
}
public void run() { // スレッド実行
  int i, j, k;
  StringBuffer strBuf = new StringBuffer(), strDon = new StringBuffer();
  boolean bend = false;

  for (;!bend;) { // ゲームループ
    try {
      Thread.sleep(50);
    } catch(InterruptedException ignore) { }

    // キー状態取得
    iKeyPrev = iKey;
    iKey = getKeypadState();

    // 終了
    if (iKey == (1 << Display.KEY_SOFT2))
      bend = true;

    if(!bOver) {
      // ポーズ・解除
      if (iKey == (1 << Display.KEY_SOFT1) && iKeyPrev != (1 << Display.KEY_SOFT1)) {
        pause(bPause ? false : true);
      }

      if (!bPause) {
        // 自機移動
        if ((iKey == (1 << Display.KEY_LEFT)) && (iMx > 0))
          iMx--;
        if ((iKey == (1 << Display.KEY_RIGHT)) && (iMx < MAPW - 2))
          iMx++;

        // 入れ替え
        swap();

        // 落下など
        if (iSeq % iSpd == 0) {
          k = fall();
          hantei();
          // 次の材料
          if (k == 0) {
            startFall();
            // ゲームオーバーか？
            for (i = 0; i < MAPW; i++) {
            if ((strMap.charAt(strPos(i, 0)) != ' ' && strMap.charAt(strPos(i, 0)) != 'E') &&
                (strMap.charAt(strPos(i, 1)) != ' ' && strMap.charAt(strPos(i, 1)) != 'E')) {
                bOver = true;
                setSoftLabel(Frame.SOFT_KEY_1, "Yes");
                setSoftLabel(Frame.SOFT_KEY_2, "No");
              }
            }
            setNext();
          }
          // 速度アップ
          if (iDon % 5 == 0 && iDon != iDonPrev && iSpd > 5)
            iSpd--;
        }
        iDonPrev = iDon;
        iSeq = (iSeq + 1) % iSpd;
      }
    } else if (bOver) {
      // リトライ
      if (iKey == (1 << Display.KEY_SOFT1)) {
        subInit();
      }
    }
    repaint();
  }
  IApplication.getCurrentApp().terminate(); // 終了
}

public void paint(Graphics g) { // 画面描画
  int iGx, iGy, iPos, i, j;
  Image img;

  g.lock();

  // 画面消去
  g.setColor(g.getColorOfName(g.WHITE));
  g.fillRect(0, 0, getWidth(), getHeight());

  // タイトル
  g.drawImage(imgTitle, 87, 98);
  // 枠
  g.setColor(g.getColorOfName(g.GRAY));
  for (i = 1; i <= MAPW; i++)
    g.drawLine(i * (TIPW + 1) + 1, 0, i * (TIPW + 1) + 1, (TIPH + 1) * MAPH - 3);
  for (i = 0; i < MAPH; i++) {
    g.setColor(g.getColorOfName((i == 0)? g.RED : g.GRAY));
    g.drawLine(1, i*(TIPH+1) + TIPH - 2, (TIPW+1) * MAPW - 1, i*(TIPH+1) + TIPH - 2);
  }

  // 自機
  g.drawImage(imgMachine, iMx * (TIPW + 1) + 2, MAPH * (TIPH + 1));
  g.drawImage(imgMachine, (iMx + 1) * (TIPW + 1) + 2, MAPH * (TIPH + 1));
  g.setColor(g.getColorOfName(g.YELLOW));
  g.fillRect(iMx * (TIPW + 1) + 2 + TIPW / 2, MAPH * (TIPH + 1) + 4, TIPW + 2, 2);

  // 材料
  for (i = 0; i < MAPH; i++) {
    for (j = 0; j < MAPW; j++) {
      iGx = j * (TIPW + 1) + 2;
      iGy = i * (TIPH + 1) - 2;
      switch (strMap.charAt(strPos(j, i))) {
      case 'D':
        g.drawImage(imgDon, iGx, iGy);
        break;
      case 'F':
        g.drawImage(imgFuta, iGx, iGy);
        break;
```


★なし。(埼玉県飯能市・いっせい17歳) ……【**影：**あっ，そう。……………。**つぐ美：**だまってちゃ，誌面の無駄づかいよ。**影：**何もないそうだから，答えようがない！**つぐ美：**影さんの自由にこのスペース使えるってことよ。**影：**そっか，ってもうスペースがない。】

```
        case 'C':
          g.drawImage(imgCow, iGx, iGy);
          break;
        case 'N':
          g.drawImage(imgNegi, iGx, iGy);
          break;
        case 'P':
          g.drawImage(imgPig, iGx, iGy);
          break;
        case 'E':
          g.drawImage(imgErased, iGx, iGy);
          strMap.setCharAt(strPos(j, i), ' ');
          break;
        }
      }
    }

    // 得点など
    g.setFont(Font.getFont(Font.SIZE_SMALL));
    g.setColor(g.getColorOfName(g.BLACK));

    g.drawString("得点", 88, 38);
    g.drawString(Integer.toString(iScore), 92, 48);

    g.drawString("牛丼", 88, 60);
    g.drawString(Integer.toString(iDon), 92, 70);
    g.drawString("個", 104, 80);

    // ゲームオーバー
    if (bOver) {
      g.setFont(Font.getFont(Font.SIZE_LARGE));
      g.setColor(g.getColorOfName(g.RED));
      g.drawString("生産ライン停止！！", 8, 44);

      g.setFont(Font.getFont(Font.SIZE_SMALL));
      g.setColor(g.getColorOfName(g.BLACK));
      g.drawString("再挑戦しますか？", 16, 64);
    }

    g.unlock(false);
  }
}
```

《リスト３》GyudonFact.jamのプログラム・リスト

```
PackageURL = gyufact.jar
AppSize = 5616
AppName = gyudonfactry
AppVer =
AppParam =
AppClass = GFApp
SPsize = 0
LastModified = Wed, 22 Aug 2001 22:01:19
KvmVer =
```


★'90年11月号に界王拳が使えるという人がいましたが，私も界王拳が使えるんですよね。大人になるって楽しい。（福岡市・前田宣弘27歳）……【つぐ美：影さん，かめはめ波が使えるって言ってたわよね。影：10年前にはね。うっ，でたな仙人！Dr.D：誰がじゃ。】

PC-G850S用



キーを連打，連打，連打！

RENDA for PC-G850S
（レンダ フォー）

かねん（18歳）

CHECKER FLAG

Dr.D：連打だけのゲームなので，もう少しリストが短くてもいいはずじゃが，チト長めじゃの〜。前半部のPRINTによる表示は省略してもいいのではないかな？また，行番号の付けかたが変な形式じゃな。このあたりもしっかり記述してほしい。中途半端なプログラミングをやっていては,プログラムの腕は向上しないゾ。
影：よ〜し，さっそくプレイだ〜。それ，連打だ，おりゃ〜!!
つぐ美：影さんダメー!! ポケコン壊したら，ポケコン投稿チェックができなくなる〜。

プログラムの内容

自称人間ベンチマークプログラムです。TABキーを200回連打するのに，どのくらいの時間が必要かを計測するものです。また，TABキーを50回連打するごとに，経過時間を記録します。

入力法と正しい計測

このプログラムはRAM DATA FILEを使っているので，RUNする前にRAM DATA FILEを作成しておきます。

テキストエディタモードで，Rキー(Rfileを選択)，Iキー(Initを選択)，FILE NAMEをSAVEDATA.DAT，FILE SIZEを128と入力してください(この作業は一度だけです)。実行はRUN MODEで，RUNと入力してください。

タイトル画面が出るので，そこでリターン・キーを押すと，名前入力の画面になります。名前を入力してリターン・キーを押すと確認のメッセージが出ます。

そのままリターン・キーを押すか，Yキーを押してからリターン・キーを押すと，簡単なルール説明になります。その後，リターン・キーを押すと計測が始まります。

正しく，かつ高速にTABキーを連打してください。連打する回数は200回です。がんばりましょう。

計測後は…

計測データを見たいときは，テキストエディタモードで，Rキー(Rfileを選択)，Lキー(Loadを選択)とキーを押して，SAVEDATA.DATにカーソルを合わせ，リターン・キーを押してください。

そして，もう一度テキストエディタモードに戻り，Eキー(Editを選択)を押します。カーソル・キーの上下で計測データを見ることができます。ちなみにRUNすると，このファイル(SAVEDATA.DAT)は初期化されます。いい計測結果が出たら読み込んだあと(Loadしたあと)テキストエディタモードに戻り，Fキー(Fileを選択)，Sキー(Saveを選択)とキーを押し，ファイル名を指定してください。

プログラムのしくみ

とくに変わったことはしていないと思います。コードは結構見にくいし，余計なこともしているかもしれません。行番号の付けかたは，あとで機能を追加できるようにしてみたんですが，なんかわかりにくくなっちゃいました。ちなみに，ほとんどのキー入力待ちがINPUTになっているのには理由があります。それは，INKEY＄によるループにすると，その間BUSY状態になり，電池の消費が気になったからです。そんなに消費しないと思いますけど，気になったので。

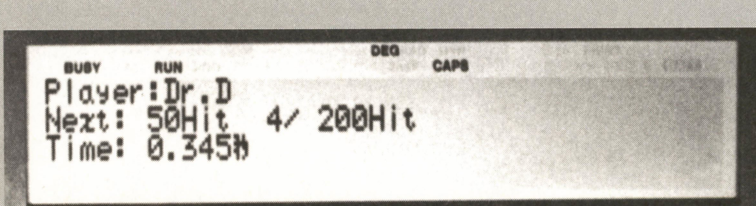
それから，経過時間ですが正しくカウントされていない場合があると思います。その場合は変数Uの値を変えてみてください。

プログラム入力法と注意点

コロン(:)とセミコロン(;)を間違わなければ大丈夫だと思います。１行が結構長くなってしまったので，気をつけて入力してください。RUNする前RAMDATA FILEの作成を忘れないでください。

改造のアドバイスと苦労した点

苦労したところは，経過時間の調整です(変数Uの数値)。なかなかいい数値が出ませんでした。キーを押すと，その処



《写真１》とにかくキーを連打するゲームです

《第１表》変数名

N＄	プレイヤーの名前
U	変数Tに加える数(この数値で秒を調整できます)
T	経過時間(単位は秒)
CT	何回目のキー入力でSAVEDATA.DATに保存するかのチェック
O	CTによる保存時の経過時間
C	キーを押した回数
MX	ゲームを終了させる数値
Z＄	仮に使っている文字変数
ZZ	仮に使っている数値変数

★今月のベスト・プログラム予想。今月はこれ↑（東京都練馬区・ルストフレム14歳）……【影：おしい！といっても，これがどのページ下に入るかは私自身もわからないんだよ。つぐ美：え？影さんわからないの？編：え？影：知らないのは私だけ？】

理に時間が取られて，もしかしたらもう少し調整したほうがいいかもしれないです。それから，キーの押しっぱなしで連打と判断しない処理に苦労しました。

改造のアドバイスは，連打するキーを変更することでしょうか。作成時はリターン・キーやスペース・キーを連打するようにしていたんですが，「一番よく使うと思われるキーを酷使するのはどうか」と思って，自分があまり使わないキーを使うようにしました。他には経過時間を記録する間隔を変更するのもいいかもしれません。

最後に

こんなプログラムでもテストプレイをしてくれた友人に感謝します。RAM DATA FILEのSAVEDATA.DATに割り当てる領域は128バイトで十分だと思います。ただし，変数MXの数値が200よりも大きい場合，128バイト以上になる場合があります。

プログラムの感想やアドバイス，入力方法で困ったことなどがあったらメールをください。アドレスは，sts-sasa@geocities.co.jp です。ホームページではVB6で作ったツールを公開しています。機会があったら見てください。アドレスは http://www.geocities.co.jp/SiliconValley-Bay/4189/index.html です。

なお，このプログラムを実行してキーが壊れても責任は持ちません。

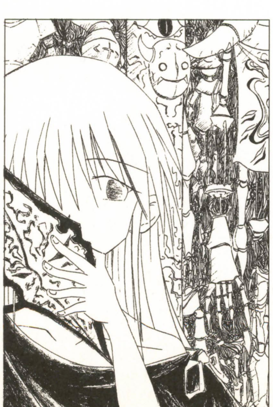
『RENDA for PC-G850S』のプログラム・リスト

```
1 'RENDA for PC-G850S
2 CLS :CLEAR
5 '---ﾀｲﾄﾙ---
6 PRINT "-- RENDA for PC-G850S --":PRINT :PRINT :PRINT "   [ﾘﾀｰﾝ]ｷｰｦ ｵｼﾃｸﾀﾞｻｲ｡":PRINT :INPUT
"Copyright(C)2001 Kanen.";Z$
10 '---ｼｮｷｾｯﾃｲ---
20 U=0.015:CT=50:MX=200
30 CLS :PRINT "<< ﾅﾏｴﾆｭｳﾘｮｸ >>":PRINT "ﾅﾏｴｦ 7ﾓｼﾞｲﾅｲﾃﾞ ﾆｭｳﾘｮｸ":PRINT "ｼﾃｸﾀﾞｻｲ｡"
40 N$="":Z$="Y":LOCATE 0,3:FOR ZZ=0 TO 23:LOCATE ZZ,3:PRINT " ";:LOCATE ZZ,4:PRINT " ";:LOCATE
ZZ,5:PRINT " ";:NEXT ZZ
50 LOCATE 0,3:INPUT "ﾅﾏｴ->";N$:IF N$="" THEN GOTO 40 ELSE PRINT "<< ｶｸﾆﾝ >>":INPUT "ｺﾚﾃﾞ
ｲｲﾃﾞｽｶ?(Y/N)";Z$:IF Z$="Y" THEN GOTO 60 ELSE GOTO 40
60 CLS :PRINT "<< ﾙｰﾙｾﾂﾒｲ >>":PRINT "[TAB]ｷｰｦ ﾚﾝﾀﾞｼﾃｸﾀﾞｻｲ｡":PRINT STR$ (MX);"ｶｲ ﾚﾝﾀﾞｽﾙﾄ
ｸﾘｱﾃﾞｽ｡":INPUT "[ﾘﾀｰﾝ]ｷｰｦ ｵｽﾄｽﾀｰﾄﾃﾞｽ｡";Z$
70 PRINT "ﾃﾞｰﾀｦ ｼﾞｭﾝﾋﾞｼﾃｲﾏｽ...":CLOSE :OPEN "E:SAVEDATA.DAT" FOR OUTPUT AS #2:PRINT #2,"":CLOSE
:OPEN "E:SAVEDATA.DAT" FOR APPEND AS #2:PRINT #2," -- ";N$;" ﾉ ｷﾛｸ --":CLS
100 '---ﾒｲﾝﾌﾟﾛｸﾞﾗﾑ---
110 GOSUB *PUTSTATUS
120 WHILE ASC ( INKEY$ ) <>10:T=T+U:GOSUB *PUTTIME:WEND
130 WHILE ASC ( INKEY$ ) =10:T=T+U:GOSUB *PUTTIME:WEND
140 C=C+1:IF C=CT THEN O=T
150 IF C>=CT THEN PRINT STR$ (CT);"Hit!!["; STR$ (O); CHR$ (247);"]      ":PRINT #2, " "; STR$
(CT);"Hit Time:"; STR$ (O); CHR$ (247):CT=CT+50:IF CT>=MX THEN CT=MX
160 IF C>=MX THEN GOSUB *PUTSTATUS:LOCATE 0,4:PRINT "--- ｹﾞｰﾑ ｸﾘｱ ---":FOR Z=0 TO 3000:NEXT
Z:LOCATE 0,4:PRINT "--- ﾏﾀﾁｮｳｾﾝ ｼﾃｸﾀﾞｻｲ ---":CLOSE :CLEAR :END
999 GOTO 100
1000 '---ｻﾌﾞﾌﾟﾛｸﾞﾗﾑ---
1001 *PUTSTATUS
1002 LOCATE 0,0:PRINT "Player:";N$:PRINT "Next:"; STR$ (CT);"Hit "; STR$ (C);"/"; STR$
(MX);"Hit  "
1009 RETURN
1010 *PUTTIME
1011 LOCATE 0,2:PRINT "Time:"; STR$ (T); CHR$ (247);"  "
1019 RETURN
```



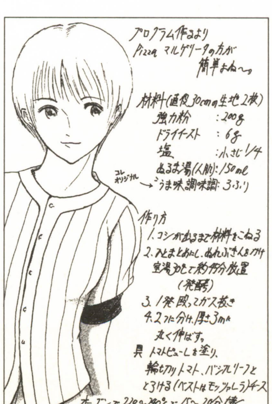
by バグニンゲン

評…ほのぼのした編集部が目に浮かびます（編）



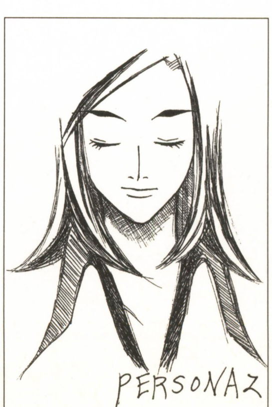
by 諏佐陽子様江

評…どうも誰かに見つめられてる気がするんだけど（影）



by 南斗ライバ

評…このレシピでできたピザを食べたいものだ（影）



by 冬虫夏草

評…噂が現実に！「噂システム」搭載だよ、影さん（編）

《今月のお楽しみプレゼント》①「トレンドマイクロ特製Tシャツ」／2名様（提供：トレンドマイクロ） ②36枚撮りネガフイルム1カートン／2名様（提供：コニカ） ③キーホルダーほか各社グッズ／10名様（提供：編集部）



CMM試聴室

毎月みなさんの曲を紹介するコーナーです。募集要項をご覧のうえ，どしどしご応募ください。ここで紹介しているデータは，CMMのWebSite（http://www.basicmagazine.net/cmm/）に収録されています。曲への意見，アドバイスなども大歓迎します。

Silent ……自作

投稿者：ナガタヒロカズ（23歳）

E-Mail：toshio@binary-id.net

制作音源　MU90B

歌モノを意識して作りました

今回は歌モノを意識して作りました。毎度のことながら，XGworksのXGエディタの操作をリアルタイム録音する機能を駆使し，インサーションエフェクトやイコライザなどを細かく設定することで音作りにこだわっています。また，ドラムパートにピッチベンドをかけることで，独特の雰囲気を出そうと試みました。

いつも曲にタイトルをつけるときは，制作中にふと浮かんだ言葉を使い，あまり意味を考えないので，私の作る曲は，しばしタイトルと曲調があってないことがあります（笑）。タイトルの「Silent」は，本来「静か」という意味ですが，曲のほうはどちらかと言うと音数も多く「騒がしい」感じになっています。

曲の終わりかたについて，適当なエンディングが思いつかなかったので，やむを得ずフェードアウトしています。なにはともあれ，お聴きください。

編集部から

メロディが全面に押し出されたポップな曲調の曲です。ユーロビートっぽさやダンスっぽさなど，最近のポップス的な響きが多いですね。音色や打ち込みのテクニックはかなりのものだと思います。聴いて気になった部分はデータをのぞいて解析してみましょう。

その打ち込みについてですが，イントロをぱっと聴いて「ふわふわした感じ」を受けました。ベースがちょっと弱めなのと，バスドラムのアタックがはっきりしないのがその理由だと思います。アタックを速めてエフェクトはドライ気味に，もしインサーションが余っている環境ならコンプレッサーをかけると，ばきばきしたバスドラムでメリハリが出てきます。つぎに曲の構成についてですが，歌モノということで，歌詞を具体的に意識してみてください。今のメロディだと，歌詞をあてはめるのが苦しくありませんか？また，いちばん聴かせたいフレーズはどれでしょう。曲を通して聴いたときに印象に残るようなフレーズを作ることに注意すると，歌モノはかなりそれっぽくなってきます。

Fazenda ……自作

投稿者：松田茂喜（49歳）

URL：http://www2.ocn.ne.jp/~somet/

制作音源　SC-8850

Sambaのリズムを取り入れたラテン曲です

Sambaのリズムを取り入れたラテン曲です。ようやく今年2曲目が完成しました。今年2回目の投稿になります。

とりあえず雰囲気を重視して細かなことには気にしていません。とくに雰囲気を出すためスモールクラブの音を入れたりかけ声を入れました。8850のほうがかけ声の種類も多いのですが，音源としてポピュラーな88Proで互換性を多く取りました。またメロディは日本の歌謡曲にも通じるものがあり，ラテン歌謡というジャンルでもいけるかもしれません。

さて，曲はパーカッション・ギター・ピアノ等のラテンリズムのうえにアコースティックギターをフィーチャーし，パンフルートをアドリブに用いています。構成は単純ですがリピートごとに少しずつ変化を持たせ，単純な繰り返しを避けたつもりです。4分強の曲にまとまっていると思いますがいかがでしょうか。

編集部から

一言で言うと「渋い」曲です。しかも，なかなかかっこよくまとまっていると思います。かけ声のボイスはクセがあって使いにくいのですが，うまく曲に取り込めていますね。また，各パートのバランスと音の抜けが抜群にいいです。SC-8850はSC-88Proよりも音の分解が良くなっているのですが，SC-88Proで聴いてもかなりの抜けの良さでした。そのミキシングですが，中央に位置する音がちょっと寂しいですね。曲のウリであるラテンパーカッションが耳ざわりに思える場面も出てきます。ある程度中央の，この曲の場合は奥のほうに，ベース系の楽器を配置すると腰がすわってきて聴きやすくなると思います。かけ声の音に関しては，ピッチを若干変えると表情が変わってきますので試してみてください。また，ギターのメロディは，もう少し大きめにしたほうがいいでしょう。フレーズ的には覚えやすいのですが，パーカッションの印象に負けていて，聴いたあとに頭に残りにくくなっています。細かいところですが，最終的に印象を大きく左右する部分なので，ぜひ調整してみてください。

●応募方法：本誌とじ込みハガキにプレゼント記入欄があります。一つだけ選んでお送りください。●締め切り：11月7日必着　《9月号当選者》①愛媛県新居浜市・西川美由紀さん他1名　②神奈川県川崎市・桜井道孝さん他4名　③福島県郡山市・近内沖尋さん他2名　おめでとうございます。

マイコンBASICマガジン

明日のスター☆プログラマー

今月のベスト・プログラマー

編：普段よりも投稿数が多かった今月ですが，そのわりに目立つ作品が少なかったように思いました。

つぐ美：確かにそうね。アイデア面でもそうですが，リスト面でも内容のわりに長いリストだったりする作品が多かったですね。

Dr.D：最近では，超大作のジャンルも定着してきているので，超大作での掲載狙いということもあるのかもしれんのう。それでもリストを短く記述していくという姿勢は，たとえアマチュア・プログラマーでも忘れてはならないゾ。完成した作品の無駄を省いていくという確認も含めてリストの見直しをじっくりしてみてはどうかな？意外といろんな無駄が発見できるハズじゃ。

編：そんなDr.のありがたい助言をもらったところで，今月の採用作品を見てみると，やはりiアプリ投稿が目に付くなぁ～。今のベーマガの中で一番投稿の勢いがあるジャンルだけに，内容もなかなかのものじゃないかな？

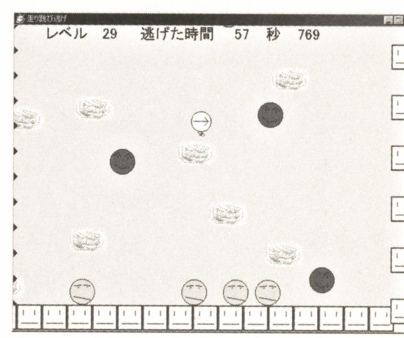
編：iアプリは，実機での容量が限られているため，自然とリストもコンパクトになりますし，そういう意味でもベーマガ向きといえるジャンルかもしれません。ただ，投稿作品を見ていると，アイデア面でもう一つ個性に欠けている印象なんですが…。

つぐ美：そのアイデア面で光っているといえば，HSP用の『走り跳び逃げ』じゃないかしら？ゲーム内容は，アクション・ゲームの基本形ではありますが，キー入力のちがいでモードをわけるというのはアクション・ゲームならではの方法だと思いました。

Dr.D：なるほど。キー入力わけのアイデアだけでなく，ゲームとしても，スピード感があり，しっかり遊べる形に仕上がっておるのぉ～。また，リストもけっして短くはないが，うまくまとめてある形じゃな。作品の内容から見ると，超大作の2作品も目に付くのじゃが，今回はリスト面での評価を重視して，このHSP用の『走り跳び逃げ』がベストにふさわしいじゃろう。

編：というわけで，今月のベスト・プログラマーは，HSP用『走り跳び逃げ』の作者・INFさんに決定です。おめでとうございます。

影：ベストに選ばれた作品は，もう読者にはおなじみ（？）の「Dr.Dのベスト・プログラム研究室」で，よりくわしく解説しているゾ。プログラム投稿者ならずとも必見だ！



▲HSP用『走り跳び逃げ』

今月の掲載者から一言

Windows（Visual Basic V6.0）用
Shino Buster（シノ　バスター）
ていこく（22歳）

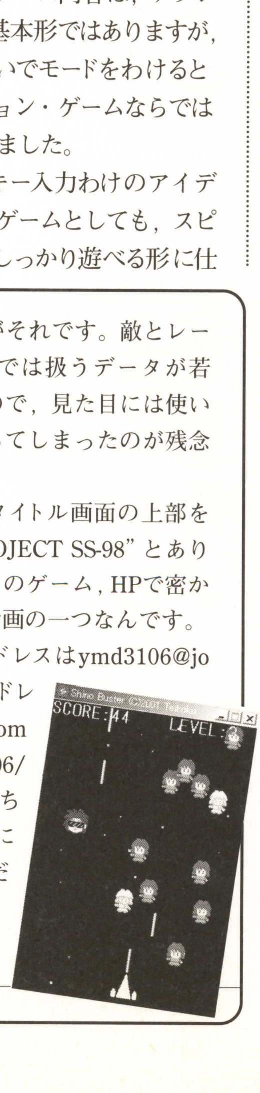
作成後記

超がつくほどオーソドックスで，しかも地味なゲームです。“Shino”というのは，私のHPのマスコットキャラをデフォルメしたものなんです。パソコンの中身を整理していたらこの絵が出てきたので，これを使ったゲームを作ってみようと思ってできたのがこのゲームです。

また，いつも使っている配列の代わりに構造体というのをはじめて使ってみました。スプライト関係のSprite型がそれです。敵とレーザーと星では扱うデータが若干異なっているので，見た目には使いにくい感じになってしまったのが残念です。

このゲーム，タイトル画面の上部を見てみると，“PROJECT SS-98”とありますが，じつはこのゲーム，HPで密かに実行している計画の一つなんです。

私のメール・アドレスはymd3106@jomon.ne.jp，HPアドレスはhttp://www.jomon.ne.jp/~ymd3106/です。感想お待ちしています。HPにも遊びに来てくださいね。



投稿ありがとう

吉久史彦／Hase／高橋好人／宮尾歩／7M4NEK／よっち／森岡拓也 with KOD／清家　巧／戸田文彰／西野紀之／YUITA／石山明弘／INF／伊藤　優／望月智則／永田洋一／福司謙一郎／三好利治／竹川友章／松本輝彦／岡崎　務／hani／橘　一馬／Mr.親父／りっちもんどかねん／三留聡／佐々木一彦／はらだ／大野孝寿／kondo／畑山隆一／森高健典／ダブル・ひろし／和田知幸／高取大典／土佐彰男／canopy／Dr.O

※全員紹介できなくてごめんなさい

投稿プログラム・コーナー

Programmer's Room

今月の総評

編：今月は投稿数が多く，投稿面ではうれしい状況でした。これからも，この勢いを持続していってください。

ただ，作品の内容面では，若干さびしい感じでした。これは，アイデアがイマイチという話でなく，以前掲載した作品のアイデアそのまま，という作品が多いという状況です。これは，このページの「一口CHECKER FLAG」を見てもらえればわかるでしょう。

とくに最近は，いわゆる「暗号化ソフト」が目に付きます。「過去に掲載されたジャンルだから，同じレベルの作品を投稿すれば掲載されるのでは？」と考えているのかもしれませんが，それではプログラム投稿者としては失格です。少なくとも掲載された作品を内容・リスト面ともに研究し，その作品の長所・短所をしっかり理解したうえで作品を作っていってほしいものです。そうすれば，多少なりとも，他の作品とのちがいが生まれてくるハズですよ。

これらを踏まえて，個性に富んだプログラムを投稿してきてほしいものです。

一口CHECKER FLAG

●Visual Basic用『25Lights』／望月智則(17歳)
影：細かな独自のアイデアはあったが，基本はライツアウトなので，チョット面白味に欠けた感じだったな。

●Visual Basic用『Mr.Encode』／7M4NEK(14歳)
Dr.D：内容は，今月掲載の『RANDOM CODE』とほぼ同じじゃが，容量の大きいファイルの変換に難アリじゃ。『RANDOM CODE』と比べると，変換速度，出力ファイルサイズともに劣っておる。リストは短めで良かったのじゃが。

●Visual Basic用『くるくる』／よっち(18歳)
影：『25Lights』同様，ライツアウト風の作品。ライトの代わりに数字を使うという変更があったけど，本質に大きな変更がないため，やはり面白味に欠くな。

●QBasic用『Yoke Yoke!! ～「丸」第二章～』／森岡拓也 with KOD(？歳)
Dr.D：障害避けゲームじゃな。ゲームとして遊べる形になっており，決して悪い内容ではないのじゃが，もう一歩味付け不足といった感じじゃな。自機の動きが典型的な2次関数の動きでは味気ないゾ。

●Visual Basic用『アナログ時計』／清家　巧(16歳)
つぐ美：タイトルどおりの基本的な内容なので，面白味に欠けるかな。いろんな設定ができる仕様だけど，どうせならイベントが発生するような機能がほしいわね。

●Visual Basic用『RUN AWAY』／戸田文彰(21歳)
影：画面はキレイにできているけど，今のままでは，いくら避けゲームと説明しても，弾発射機能がない欠陥シューティングという印象。まずは，シューティングとしてゲームを完成させよう。

●Visual Basic＋DirectX用『Super Shoot』／西野紀之(14歳)
つぐ美：敵が無造作に出てくる形式のため，すぐに飽きてしまいます。また，敵キャラはオリジナルのものでお願いね。

●F-BASIC用『MuCom』／伊藤　優(18歳)
Dr.D：MIDIファイルを出力できるのはなかなかじゃが，肝心の簡易作曲画面のマス目が細かくて見にくいのぉ。これなら五線譜上で作曲したほうが楽そうじゃ。

●VisualC++用『ボイス・チェンジャー』／福司謙一郎(16歳)
編：技術的な面は，評価できる作品なのですが，肝心の変換される声が，あまりきれいに再生されないのは痛いですね。

●VisualC++用『ランダムドットステレオグラム』／永田洋一(23歳)
つぐ美：最近，視力回復に良いと言われているステレオグラムの作成ソフト。Windows環境を生かして，自分が描いた絵だけでなく，bmpファイルから作成できるといった機能があるといいのでは？

●HSP用『神経衰弱 あわせてポン』／三好利治(46歳)
つぐ美：オーソドックスな内容ながらも，ゲームとしてはしっかり作られていました。ただ，画像ファイルなしでも遊べると良かったのですが。また，画像はオリジナルなものにしてください。今のままでは，著作権の面でも採用できません。

●PC-G850用『ポケコンカードバトル』／竹川友章(20歳)
Dr.D：凝ったゲーム設定じゃが，リストがチト長いのぉ～。最近では，あまりリストの長さを問題視していなかったが，少なくとも，入力して遊ぶ人が多いと思われるポケコンでは，リストの長さにも気を配ってみてほしい。

今月はもう一歩でした

●Visual Basic用『火山』／宮尾　歩(13歳)
編：アクション・ゲームとしては，コンパクトに作られており，なかなか良いのですが，すぐにゲーム・オーバーになりやすいため，プレイ時間が短くなりがちなのが残念。ゲーム面でもう一工夫あると良いのですが。



◀ていこく

Dr.Dの LABORATORY ベスト・プログラム研究室



今月のベスト・プログラム：HSP用『走り跳び逃げ』(66ページ掲載)

今月のベスト・プログラムは，キー入力によってモードを分けているのが特徴のジャンプ・アクション・ゲーム，HSP用『走り跳び逃げ』じゃ。単純なゲーム内容ながらも，アクション・ゲームとしての基本的な処理をほとんど網羅していると言えるこの作品。その点を重視して，今回は特徴であるキー入力の他，リスト全体の流れについても解説していこう。

(Dr.D)

キー入力処理

キー入力処理については，この作品のCHECKER FLAGでも触れておるが，ここでは，さらにくわしい解説をしていこう。

この作品は，リストを誌面に掲載しているが，解説を入れた形式でキー入力に関連する部分を**リスト1**に掲載してみた。

この部分の核になるのは，HSPならではの“stick”命令じゃな。この命令はキー入力を得るだけでなく，非トリガータイプキー指定ができるのが特徴じゃ。この機能を使うことにより，押しっぱなしの間もキー情報を得ることが可能になるのじゃ。“五段跳び式”モードでは，↑キーを連射する形式のため，↑キーの非トリガータイプキー指定は行わない形になる(stickの第2引数が，←→キーのみを非トリガーに設定するため，5になる)。このゲームのモード分けは，“stick”命令があったからこそ，手軽な形式で実現できたと言えるな。

では，この部分を踏まえたうえで，各モードの処理のちがい(＝ジャンプ部分のちがい)をそれぞれ解説していこう。ちなみに，↑キーの入力判定は“if key&2”となっている部分じゃゾ。

まずは五段跳び式から。このモードは，↑キーを押すごとにジャンプの処理が行われる。jpspという変数を用いて5段階までの判定を行い，5段階以内ならば，snd 0でwavを再生し，jumpowの値を設定してジャンプを行うようにしておる。

つぎは下方推進式モード。このモードのジャンプ処理は，ただ，jumpowの値を設定しているだけじゃが，このままでは，画面上までジャンプし続けてしまう。それを防ぐため，*itsmeルーチンにあるif文を用いて，画面上まで行ったら，下方向へ落下するように設定してある(jumpow=60の部分)。

また，このモードはジャンプの速度が，だんだんと上がっていくが，これはjumpowの値をjumpow-=8いう形(これはjumpow=jumpow-8と同じ意味)で設

≪リスト1≫キー入力部分の解説リスト

```
*itsme;------------ 自キャラの処理
 途中省略
        if (mey<0)&(mode=1) : mey=0 : jumpow=60
        ※ここで，下方推進式モード時の落下処理の設定を行う
 以下省略

 *keystatefive;---------- キーの処理(五段跳び式)
        stick key,5  ※キー入力情報を得る
        if key&1 : mevx-=12  ※←キーの処理
        if key&4 : mevx+=12  ※→キーの処理
        if (key&2) : if (jpsp!5) : snd 0 : jumpow=-26 : jpsp++
        ※ジャンプ部分の処理
        ※if (jpsp!5)があるのは，五段跳びを行うため
        return
 *keystgrav;---------- キーの処理(下方推進式)
        stick key,7  ※キー入力情報を得る
                     ※スペース キーも非トリガー設定
        if key&1 : mevx-=12
        if key&4 : mevx+=12
        if (key&2) : jumpow-=8
        return
*keystplus;---------- キーの処理(可変加圧式)
        stick key,7
        if key&1 : mevx-=12
        if key&4 : mevx+=12
        if key&2 {
                ※ジャンプ処理。
                ※if文があるのは，画面上以上に移動するのを防ぐため
                if jpsp!10 : jumpow=-30 : jpsp++
                if jpsp=1 : snd 0
                } else : jpsp=10
```

定しているため，だんだんとジャンプの速度が上がっていくというわけじゃ。

最後に可変加圧式モード。このモードの処理も基本的には，下方推進式モードと一緒じゃ。ただ，jumpowの値の設定方法が，下方推進式モードとはちがい，-30という値を常に代入する形式のため，キーを押している間は一定の速度でジャンプするというわけじゃ。

また，if jpsp!10の処理は画面上までジャンプし続けてしまう現象を防ぐためのもの，if jpsp=1の処理はwavを再生するためのものじゃ。

リスト全体の流れ

タイトルなどを除いたゲーム中の処理は，67行目からのrepeat文に囲まれたgosubが多く記述されている部分で実現しておる。この部分の各ルーチンの処理内容を**第1表**にまとめてみた。

ゲームを作るにあたっては，このように各処理別にルーチンを作成しておいて，最後にそれらを組み合わせるという作成方法が，いわば王道と言えるな。

このような作りかたをすることで，各ルーチンごとにプログラムを作成できたり，場合によっては敵のルーチン部分のみを差し替えることで，簡単に敵の動きを変更する，といったことが可能になるゾ。

改造してみよう

今回の改造は，CHECKER FLAGでつぐ美クンが述べていた，レベルアップによるスピードの初期化をなくしたバージョンの作成と，各変数の値の変更などによる，簡単な改造を紹介していこう。

まずは，スピードの初期化をなくしたバージョンの作成。これは*timersetcルーチンにあるレベルアップ処理の部分を改造すればOKじゃ。ただ，このままスクロール・スピードを増加させつづけると，自キャラの移動速度よりも，画面のスクロール・スピードのほうが上回ってしまい，ゲームにならない。

というわけで，スペース・キーによるターボ移動（と勝手に命名）を付けてみたゾ。これらの変更点は，**リスト2**にまとめてみた。

つぎに，変数の値などを変更する簡単な改造。これは**第2表**にまとめておいたので，ぜひとも試してみてほしい。

最後に

今回の作品は，HSPで作られていることもあり，こういった変数の変更による改造が行いやすいハズじゃ。

ぜひともこういったプログラム改造を経験して，作品や言語の理解をより深めてほしいものじゃ。

≪第1表≫各処理ルーチン表

ルーチン名	処理内容
*mapdraw	背景を描画。スクロール処理もここで行う
*timersetc	経過時間処理＆その表示。レベルアップ処理もここで行う
*itsme	自キャラのキー入力＆変数処理。描画は行わない
*enehenyo	敵１(地面の敵)の設定＆描画
*enetorya	敵２(跳ねている敵)の設定＆描画。移動処理もここで行う
*medraw	自キャラの描画
*hitcheck	当たり判定を行う

※*medrawと*hitcheckの間にあるif文は，ゲーム・オーバー判定を行う
await waitsはウェイト処理

≪第2表≫変数変更による改造例一覧

改造箇所	改造記述例	効果
enemy=0(255行目)	enemy=5	最初から現れる敵の数を増加
speed=2(259行目)	speed=5	最初からスクロールスピードを上げる
mevx-=12など(145行目 他)	mevx-=24	自キャラの移動速度を上げる
jumpow=-26など(147行目 他)	jumpow=-39	ジャンプ力の増加
if (jpsp!5)(147行目)	if (jpsp!8)	ジャンプの段階の増加(五段跳び式のみ)

≪リスト2≫スクロールとターボ移動の改造リスト

```
;スクロール部分の改造
mes "レベルアップ"
if timemilli<waits {
        level++
        ;改良点
        ;変更前 ;if (level¥8=1) : enemy++ : speed=2 : else : speed+=2
        ;変更後。speedの初期化(speed=2)をなくした形になる
        if (level¥8=1) : enemy++
        speed++
}


;ターボ移動の改造部分
stick key,5
if key&1 : mevx-=12
if key&4 : mevx+=12
if (key&2) : if (jpsp!10) : snd 0 : jumpow=-40 : jpsp++
;変更(ターボ移動処理。36の値を変更することで，より大きな移動も可能)
if key&16 : mevx+=36
return
```

PlayStation 2でゲームが作れる！



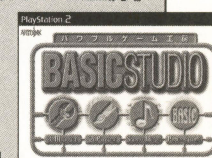
「BASIC STUDIO パワフルゲーム工房」入門

第7回
3Dモデルを使って迷路を作る

文／山崎　寛

「BASIC STUDIO」の特徴として，3Dモデルをかなり簡単に扱えるという点があげられます。前回のサンプルでは基本的にカメラを固定した3D表示でした。カメラを動かすとダイナミックに画面を変化させることができるので，今回はカメラを動かして「3D迷路」を作ってみましょう。

『BASIC STUDIO パワフルゲーム工房』
発　売：アートディンク
対応機種：PlayStation 2
メディア：CD-ROM 2枚組
　　　　　発売中
価　格：9,800円（税抜）
http://www.artdink.co.jp/japanese/title/bs/index.html

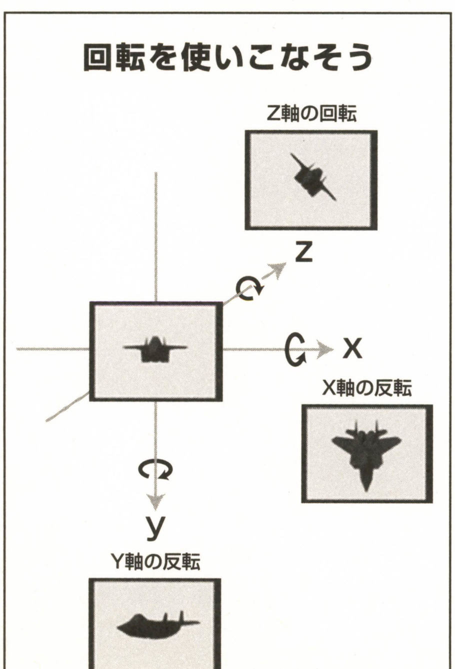


3Dの回転

3Dモデルを扱う際，何をするにしても避けられないのが回転です。

キャラクターなどを好きな方向に向けたり，カメラ（視点）を動かしたりと，非常に回転が重要な要素を占めているからです。

回転を指定する場合，**第1図**のようにX軸，Y軸，Z軸の三つの要素それぞれについてどのくらい回転させるのかを指示します。どの軸を回転させれば自分が思うとおりの回転をするのか，しっかりと頭に入れておきましょう。また，BASIC STUDIOでは角度の指定に「ラジアン」という単位を使います。通常私たちが使っている「ディグリー（度）」という単位では円の1周を360とした値ですが，ラジアンでは1周を2πとした値となっていることに注意してください（**第2図**）。



《第1図》どのくらい回転させればいいのか，考えながら指示してみよう！

3D迷路を作る

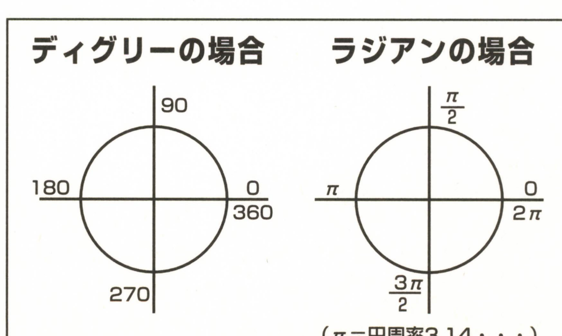
今回のサンプルとして，3D迷路を作ってみることにしました。

3D迷路といえば，非常に古典的で，昔はいかに3D表示するかに苦労してワイヤフレーム表示などでプログラムしていたわけですが，さすがに3Dモデルを簡単に扱えるBASIC STUDIOでは，苦労することなく簡単に作ることができます。

方法としてはとても単純で，迷路の形に四角い3Dモデルを並べ，プレイヤーのいる場所，プレイヤーの向いている方向へとカメラを常に動かすだけです（**第3図**）。

迷路は，配列変数にブロックの有無を入れ，壁との衝突判定もこれを利用しています。まあ，常套手段といえるでしょう。

操作はパッドの左右で方向転換，上で前進，下で後退という一般的なものにしています。動作がドット単位なので，それなりにスムーズに動きますよ（**写真1**）。

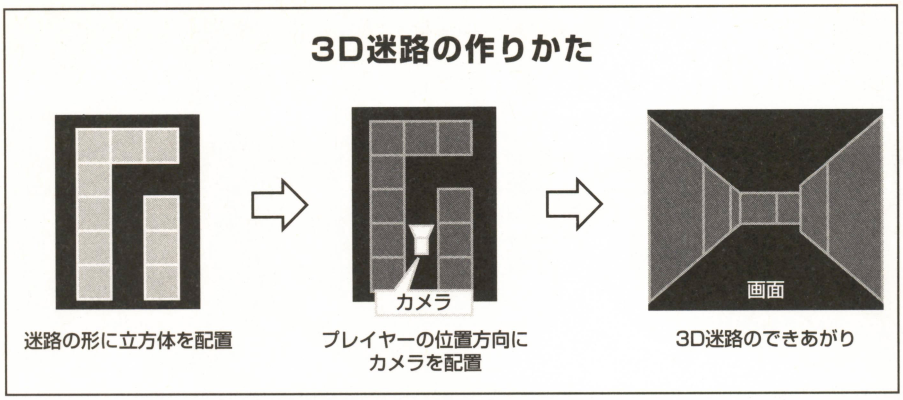


《第2図》BASIC STUDIOでの角度指定はディグリー（度）ではなく，ラジアンで指定する

迷路を自動作成してみる

3D迷路も，常に決まった内容ではつまらないので，ゲームをするたびに迷路の内容が変わるようにしてみましょう。つまり，プログラムで迷路を自動的に作るのです。

こうしたプログラムを考えるのは，パズルのように頭を使うという，プログラムの一番基本的な面白さとも言えるアルゴリズムの楽しさを味わえます。迷路作成はかなり古典的なテーマなので，プログラムの経験のある人なら一度くらいは試したことがあるで



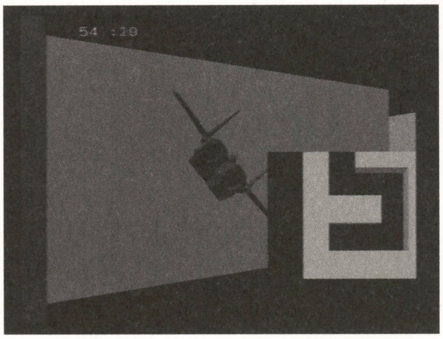
《第3図》このような手順で簡単な3D迷路ができあがるのだ

しょう。

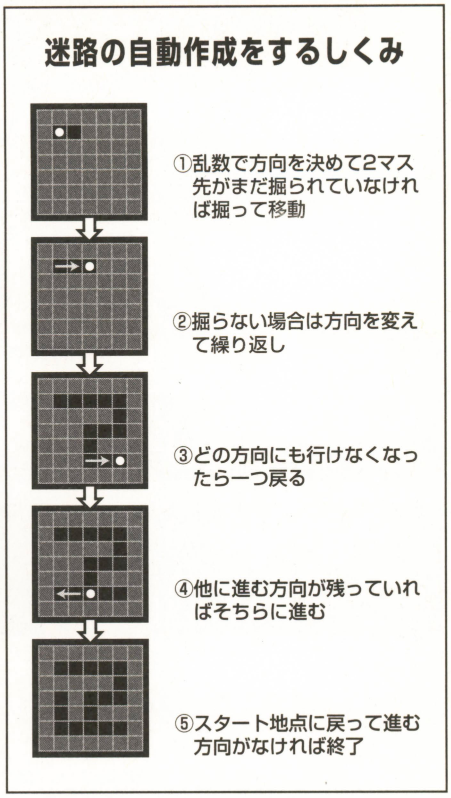
今回のプログラムでは「穴掘り法」とも言えるアルゴリズムで迷路を作成しています（第4図）。

このプログラムで毎回異なる迷路になるのは，930行のRANDOMIZE文で乱数系列を不定に初期化しているためです。

RANDOMIZE文の後に一定の数値を指定した場合，指定した数値によって一定の迷路を作成するようになります。



《写真1》サンプル・プログラム実行画面



《第4図》穴を掘るように迷路を作っていく

《リスト1》サンプルゲームのプログラム・リスト

```
100 !●3D迷路のサンプル
110 CALL SYOKI
120 DO
130   CALL IDOU
140   CALL HYOUJI
150   FSYNC 1
160 LOOP
170 !●初期化
180 SUB SYOKI
190 INIT
200 CALL MAPSET
210 !
220 SCREEN WINDOW 400,100,639,223 ―― マップ用子画面表示
230 CAMROT 2,PI/2,0,0 ―― カメラ2(子画面)の設定
240 MODSET 1,7,5 ―― 飛行機をモデル1に設定
250 MODOPT 1,0.5,1,1,1 ―― モデル1の大きさを1/2に
260 MODDISP 1
270 MODROT 1,0,PI,0
280 FOR J=1 TO 21
290   FOR I=1 TO 21
300     IF MAP(I,J)>0 THEN
310       MODSET J*21+I,2,MAP(I,J)
320       MODDISP J*21+I
330     END IF
340     MODPOS J*21+I,I*40,0,J*40
350   NEXT I
360 NEXT J
    (280～360: マップの内容に合わせて立方体を並べて表示)
370 !変数初期化
380 MX=120
390 MZ=120
    (380～390: プレイヤーの位置)
400 MR=PI ―― 向いている方向
410 MRL=0 ―― 飛行機のかたむいた角度
420 MSP=3 ―― スピード
430 TIM=0 ―― 時間カウント用
440 CAMROT 1,PI/2,0,0
450 END SUB
460 !●移動ルーチン
470 SUB IDOU
480 TX=MX
490 TZ=MZ
500 IF PAD(1,4)>0 THEN
510   MX=MX+SIN(MR)*MSP
520   IF MAP(MX/40,MZ/40)=1 THEN MX=TX
530   MZ=MZ+COS(MR)*MSP
540   IF MAP(MX/40,MZ/40)=1 THEN MZ=TZ
550 END IF
    (500～550: 前進)
560 IF PAD(1,3)>0 THEN
570   MX=MX-SIN(MR)*MSP
580   IF MAP(MX/40,MZ/40)=1 THEN MX=TX
590   MZ=MZ-COS(MR)*MSP
600   IF MAP(MX/40,MZ/40)=1 THEN MZ=TZ
610 END IF
    (560～610: 後退)
620 IF MAP(MX/40,MZ/40)=2 THEN
630   LOCATE 10,11
640   PRINT "GOAL"
650   END
660 END IF
    (620～660: ゴール)
670 IF MAP(MX/40,MZ/40)>0 THEN
680   MX=TX
690   MZ=TZ
700 END IF
    (670～700: カベに当たった場合)
710 IF PAD(1,1)>0 THEN
720   MR=MR+0.05
730   IF MRL>-PI/2 THEN MRL=MRL-0.1
740 END IF
    (710～740: 右回転)
750 IF PAD(1,2)>0 THEN
760   MR=MR-0.05
770   IF MRL<PI/2 THEN MRL=MRL+0.1
780 END IF
    (750～780: 左回転)
790 MODPOS 1,MX,0,MZ
800 MODROT 1,0,MR,MRL
    (790～800: 飛行機の位置，向きを設定)
810 CAMPOS 1,MX-SIN(MR+PI)*40,0,MZ-COS(MR+PI)*40
820 CAMROT 1,0,PI-MR,0
    (810～820: カメラ1の位置，向きを飛行機の後ろに設定)
830 CAMPOS 2,MX,-500,MZ ―― カメラ2の位置設定
840 END SUB
850 !●時間表示
860 SUB HYOUJI
870 LOCATE 5,0
880 PRINT INT(TIM/60);":";MOD(TIM,60)
890 TIM=TIM+1
900 END SUB
905 !●迷路作成
910 SUB MAPSET
920 DIM MAP(21,21),DX(4),DY(4)
930 RANDOMIZE ―― 乱数を初期化
940 FOR J=2 TO 20
950   FOR I=2 TO 20
960     MAP(I,J)=1
970   NEXT I
980 NEXT J
    (940～980: マップ内をうめる)
990 FOR I=1 TO 4
1000   READ DX(I),DY(I)
1010 NEXT I
1020 DATA 1,0,0,-1,-1,0,0,1
    (990～1020: 数値と方向の変換用テーブルを用意)
1030 !
1040 X=3
1050 Y=3
1060 FL=1 ―― 最初に着いた行き止まりを判定するフラグ
1070 !
1080 MAP(X,Y)=0
1090 CALL HORI(1,1)
1100 END SUB
1110 SUB HORI(H,TH)
1120 IF MAP(X+DX(H)*2,Y+DY(H)*2)=1 THEN
1130   MAP(X+DX(H),Y+DY(H))=0
1140   MAP(X+DX(H)*2,Y+DY(H)*2)=0
1150   X=X+DX(H)*2
1160   Y=Y+DY(H)*2
    (1130～1160: 2マス先がうまっていれば，穴を掘って移動)
1170   T=INT(RND*4)+1
1180   CALL HORI((T),(T))
    (1170～1180: 新しい方向を決めて掘り進む)
1190   X=X-DX(H)*2
1200   Y=Y-DY(H)*2
    (1190～1200: 一つ戻る)
1210 END IF
1220 H=MOD(H,4)+1 ―― 別の方向にする
1230 IF H<>TH THEN CALL HORI((H),(TH))
1240 IF FL=1 THEN
1250   MAP(X,Y)=2
1260   FL=0
1270 END IF
    (1240～1270: 最初の行き止まりをゴールにする)
1280 END SUB
```

Catlikeのもっと身近にプログラミング！

第5話 シューティング・ゲーム編 その3 敵艦監督を楽しもう

ゲームを作るのって，プレイヤーとの戦いだったりもするわけですよね。自分の用意した精鋭を，いかにプレイヤーに気持ちよく戦わせるか。さぁ，あなたもシューティング・プログラムで最強提督を目指しましょう！

ゲーム性時代，復活の予感！

ぼくらゲーム世代が必ずお世話になっていると言っても過言ではない任天堂が，満を持して新ゲーム機「ニンテンドーゲームキューブ」を発売しました。とくに国外からの注目度が高く，電気街・秋葉原では，発売当日，徹夜で並んでいた人の先頭は揃いも揃って海外からやってきたゲームファンとのこと。思わず，日本は平和だなぁ，なんてつぶやいたりして。それにしてもこの原稿，いつ書いているかバレバレですね。あんな危険なことがあったのに。あまり突っ込むと検閲に引っかかって編さんの「没！」の呪文が発動されそうなのでこのあたりでやめときましょうか。



さて，ゲームキューブ本体も非常にコンパクトかつキュートで魅力的なのですが，それよりうれしいのが，シンプルなゲーム性を容認してくれる市場が生まれたことでしょう。最近のゲームはとにかく複雑怪奇，ニュータイプでないと操作できないようなゲームまで世にはびこっています。

今回の新ハード投入によって，アイデアは斬新，かつ動かしてて気持ちいいゲームがいっぱい生まれてくることを期待しちゃいます。

シナリオ？ 台本？ シューティングでは？

そういう意味では，シューティング・ゲームって，基本的にはシンプルの極地！ のような気がしませんか？ でもまぁ，最近は派手になっていますし，ゲーム中にいきなり物語を彷彿とさせるイベントが挿入されたりと，あんまりシンプルじゃない気もしますね。

で。シューティング・ゲームのシナリオというと，どういうものを思い浮かべるのでしょうか。最近のゲームの大多数にはシナリオらしきシナリオが入っていますよね。ステージ間デモであったり，劇中劇だったり。けれど，そのシナリオというのは，きっと「登場人物がしゃべる」だとか，「ミッション結果の報告」だとか，「つぎのミッションの説明」などなど，ドラマが展開するようなシナリオだったりするのではないでしょうか。

筆者紹介

catlikeホームページ
http://catlike.marchen.to/
第1話がなかなか出せない「いつか，何処かで。」今月はアップできているのでしょうか…。

文:まかべひろし
年末発売する安価小型コンピュータ「P/ECE」にちょっとお熱をあげてます。フリーでできるプログラム環境もあるそうで，あわよくば，ベーマガで紹介記事を書きたいな，なんて思っていたりして…。みなさん，どうです？

絵:桜部のとす
引っ越し準備でタイヘーン。こんなにいっぱい本あったのか……。昔の絵とか発掘して「うわ〜」って言ってますよ〜(^^;

ここでいうシューティング・ゲームのシナリオっていうのは，それらとはまったく別。いうなれば作戦指令書とでも申しましょうか。プレイヤーである主人公機は，単身で敵陣に突っ込んでいき，大波のように寄せてくる敵の大群をばったばったと倒していくわけです。それがまるで時代劇の殺陣のようで非常に快感であり，シューティング・ゲームの肝ともいうべき部分なのですが，この「寄せてくる敵群」が適量でないと，気持ちいいゲームにはならないのです。

その敵軍を配置する指令書がシナリオってわけ。どのタイミングでどの部隊をプレイヤーへの障害として向かわせるか，という部分をコントロールしてあげなきゃ，ゲームが成立しないのですよ。

そこで今回は，このコントロールの手法について考えてみたいと思います。題名にあるように，敵艦の監督を演じてみ

ようではありませんか。

もちろん，ゲームを作るぼくらの最終目的は「プレイヤーにゲームを楽しませること」。普通の敵艦監督のように「プレイヤーを倒すこと」ではありません。それこそ，ただ倒してしまうだけなら，通り抜ける隙間もないほどの艦隊をプレイヤー機に向かわせてやればいいわけですよね。避ける隙間もないほどの弾幕を，画面いっぱいに張り巡らせてやれば，たちまちプレイヤー機は白旗をあげてしまうことでしょう。けど，それって，すでにゲームじゃない気がします。

もちろんプレイヤーによってゲームを遊ぶレベルはちがうわけですから，何度も何度もなーんどもテストプレイを繰り返して，難易度調整を行ってやる必要があります。そのためには，できるだけ簡単に変更が行えるシステムを作ってやる必要があるでしょう。

すべてをプログラム・コードで書いても実現は可能なのですが，小さな手直しでも大きな手直しでも，毎回書き直してやらなくてはいけないわけで，それを考えただけでも調整が面倒になってしまいます。調整が面倒じゃ，テストプレイにだって力が入りませんよね。

というわけで，それらの条件を満たすような手法を考えてみることにしましょう。例によって，これを読みながら頭の中ではシューティング・ゲームの画面をイメージしていただくわけですが，今回は『グラディウス』や『スターフォース』，最近では，トレジャー社の名作『レディアントシルバーガン』などを想像するといいと思います。え，ぜんぜん最近の例じゃないって？　ほらまぁ，そう硬いこと言わずに，ね。

でも,何を用意すればいいのさ?

と，その前に。いつもの進行手順でなんですが，実際にゲームを表現するのに，シナリオ進行として何が必要か洗い出しておきましょう。

プログラムを組むうえで一番大事なのは，考えること。モニタとキーボードを目の前に，プログラム言語を操ることよりも，やりたいことを実現するためには何が必要か，必要になったものをどう処理すればいいかを考えるのが一番大事なのですね。

ここでいうシナリオというのは，敵機を「いつ」，「どこで」，「どの機体を」，「どのように」出現させるかという指定書になります。この四つの項目を用意することになるわけですが，この中で，どの項目が一番「基準」としてふさわしいでしょうか。これは，ゲームの進行と照らし合わせて考えるとわかりやすいでしょう。シューティング・ゲームの場合，強制的に画面がスクロールしていくものが多いのですが，この「強制的に」というのがポイントです。画面のスクロール度合いは，ユーザーのプレイに左右されず，どんな人がプレイしても同じ時間プレイしていれば同じ場所にいるはずなのです。あ，もちろん，ボス戦などでスクロールが止まらなければ，ですよ。

ゲーム中，強制スクロールを組むのはそうむずかしいことではありません。1/60秒とか，1/30秒などでスクロール値をカウントアップさせていけばいいだけのことです。時間軸とも同期しているので，ゲームの進行という点から考えても非常に扱いやすい値であるといえるでしょう。これは，「いつ」および「どこで」という項目にあたります。もちろん，強制スクロールをしないのであれば「いつ」と「どこで」は別項目になるでしょう。しかしシューティング・ゲームという性質上，この2項目は同一として考えても差し支えはないと思います。そして，この値を軸にシナリオを進めていくといいという結論に達します。

残るは「どの機体を」，「どのように」出現させるかということです。「どの機体を」というのは，言葉からすぐにイメージできるでしょう。一言で敵機といっても，その種類はさまざまです。弾を撃たずに一撃で倒される貧弱な機体もあれば，ワープ航法をたくみに用いてプレイヤーをかく乱するような機体があっていいと思います。いわゆる「雑魚敵」と呼ばれるものですが，これらをバラエティに富んだものにすることで，ゲームの性格や方向性，もちろん，おもしろさだって左右されるのです。

そして「どのように」出現させるか。この項目には「何体出すか」という数は



含まれていません。敵機は基本的に一つのオブジェクトとして扱います。「いつ・どこで」「どの機体を」「どのように」出現させるかという項目は，一つのオブジェクトに一組ずつ，それぞれ定義してやる必要があります。ですから「どのように」出現させるか，という項目に，編隊を組んで出現させる，というような情報は含まれません。

ここでの情報は，「画面のどの位置から，どこに向かうような動きで」出現させるかというものに他なりません。画面外からフレームインしてくるのが普通だと思いますが，ワープ航法を用いる敵機なら，エフェクトとともに画面内に突然現れて，プレイヤーを攻撃してきたっていいでしょう。また，敵機は画面内に現れたらそこで止まってしまうわけではありません。神風特攻部隊のごとくプレイヤー機に一直線に飛んできたり，はたまたヒット＆アウェイを駆使してきたりすると，プレイヤーもその対処に困るのではないでしょうか。

というところで，とりあえず必要な情報は，出現時期を表す「いつ・どこで」，出現機体の種類を表す「どの機体を」，出現させる場所とその後の行動を決める「どのように」の3種類になると思います。

実際の定義を交えて紹介!

長々と講釈してきましたが，ここまでくれば実際にどのようなプログラムを書けばよいのか，おぼろげに見えてきているのではないでしょうか。

その一例として，すべての処理を数値データとして持つ方法を紹介しましょう。BASICではDATA文のかたまり，C言語では配列テーブルと考えてもらえれ

《リスト1》シナリオデータの定義（C言語）

```
typedef struct {
      int         nPutTime;         // 出現開始する時間
      int         nCharaType;       // 機体の種類
      int         nPutX, nPutY;     // 出現開始する位置 x,y
      int         nMoveType;        // 移動方法の種類} SCENARIOOBJECT;

// ソースの中に定義するときはこんな感じです
SCENARIOOBJECT pSnObj[] = {
      { 1000, 1, 640, 480, 1 },
      { 1010, 1, 640, 480, 1 },
};
```

《リスト2》シナリオデータ読み込みサブ・ルーチン（C言語）

```
// nTimeCntにはゲームの進行カウンタを入れてください
void ScenarioRead( int nTimeCnt )
{
      static int nSnCnt = 0;
      while( pScObj[ nSnCnt ].nPutTime <= nTimeCnt )
      {     // ここの中に定義用プログラムを書きます
            // 前回の「オブジェクト」を作るようにするといいでしょう
            // 出現場所は( pScObj[ nSnCnt ].nPutX, pScObj[ nSnCnt ].nPutY )です

            // 作り終わったら、シナリオ構造体用カウンタを＋１します。
            nSnCnt++;
      }
}
```

ばイメージしやすいと思います。各データは「いつ・どこで」「どの機体を」「どのように」出現させるかのデータで1セットにして作ります。また，時間軸をベースにして処理していきますので，「いつ・どこで」の順番に並べておくと処理が簡単になります。順番に並べるのは手作業になりますので，面倒な人は，データを読み込んだ後に時間軸順にソートを行うプログラムを組んでもいいかもしれません。

しかし，実際ゲームを作っていく過程で，データはできるだけ順番どおりに並べておいたほうが修正作業がしやすいという利点もあります。このあたりは，自分の手法に沿って作ればいいと思います。

構造体を使える言語なら，1オブジェクト分を構造体定義してしまえば，さらに簡単に組むことができます。実際，シューティング・ゲームに必要最低限なシナリオ用オブジェクト情報は，**リスト1**のような感じになるでしょう。

「どのように」の部分だけ，パラメータがnPutX,nPutY,nMoveTypeと分かれてしまいましたが，「いつ・どこで」「どの機体を」は1対1で定義できました。そして，**リスト2**のようなサブ・ルーチンを，進行カウンタがカウントアップされるたびに呼べばいいことになります。

ゲーム中の進行カウンタをカウントアップするタイミングは，背景をスクロールさせるぐらいにしておけばよいでしょう。カウントアップするついでに，**リスト2**のようなサブ・ルーチンを呼び出すようにしておきます。ルーチン内では，進行カウンタとシナリオの「いつ・どこで」の値を見比べ，進行カウンタの値以下であれば，新たな敵機を作り出します。この場合の「敵機」は，前号でお話した「オブジェクト」を一つ追加するのだと考えれば簡単だと思います。

新たに作りだすオブジェクト用の情報は，シナリオ・データに書き込んでおきます。ここでは，機体の種類，初期座標，移動方法の種類を用意しておきました。機体の種類に，画像やダメージ値等のデータがリンクされているとすれば，シナリオ・データ的にはこれだけ用意すれば十分でしょう。

リスト1のデータの例ですと，進行カウンタが1000のときに，機体タイプ1の敵機が（640,480）に現れ，移動パターン1に準じて移動していくというものになります。続くデータは，進行カウンタが1010のときに実行されるに違いありません。ちなみに，この例のように時間をずらして同じようなデータを並べておくと，編隊を組んで飛んでくる，という処理を作ることができます。敵機と敵機の間の距離は，それぞれの「いつ・どこで」のカウンタの幅で調整することができるので，計算では出しにくい「統制の取れていない」編隊というのを作るのも簡単だったりしますね。

前号のオブジェクトを用いたプログラムと組み合わせることで，シナリオ部分は「敵機を出現させるまで」実際画面に出たあとの管理はすべてオブジェクト側に任せることができます。まさに，選手を舞台に送り込んだ後は自分が下した判断が正しいことを信じて成果を待つ監督の仕事と同じと言えるでしょう。

あとは，ステージが変わるごとにゲームの進行カウンタを初期化し，かつ，参照するシナリオ構造体配列も変更すれば，ステージの構成もまとめて定義することができるようになります。

これで面クリア型シューティングもオ

ールオッケイなのです。

演出の管理はどうするの?

さて，敵機の出現スケジュールがこれで出来上がり，出現した後の処理も前号で軽く触れましたので，一応ゲーム部分に必要な情報はこれで見通しがついたことになります。真っ黒な画面の上にぱらぱらと敵がやってきて，弾を撃ってプレイヤー機を狙う，くらいのプログラムはなんとかなるといったところでしょう。

もちろん，背景は臨場感を出すために，ループ・スクロールくらいしたほうがいいでしょう。できれば，RPGやアクション・ゲームのマップチップのように，起伏に富んだ背景がゲームを盛り上げることは言うまでもありませんが，それでも，実際に遊んでみると，なんだかモノ足りないなぁと思えてしまいます。そう，派手な演出がプログラムされていないのです。それもそのはず，今の考えかたでは，演出管理自体をサポートしていないので当然です。

では，演出というのはどういうものが考えられるでしょう。一例をあげるとするならば，絶対ほしいのは敵機の爆発，もしくはそれに準じた演出です。弾が当たった瞬間，被弾した敵機がいきなり画面から消えてしまっては寂しいですし，何より撃ち落としたという爽快感がないじゃないですか。ということで，敵機の爆発演出は必須だと思うのです。

とはいえ，たかが爆発一つのために新たにプログラムを組むのもばかばかしい話です。もちろん，爆発の映像をすべて計算で作り出すんだ，となってしまえばプログラムを組まざるを得ない状況といえますが，すべての画像は画像データとして持っておくのが一般的な組みかたでしょう。つまり，爆発の絵も各オブジェクトと同様，画像データとして持っておき，画面には転送で表示するのです。まぁ，当たり前といえば当たり前ですね。

で，組みかたです。これは非常に簡単に処理することができます。まず，各敵機と同様に，「爆発」のオブジェクトを作ります。敵機種類が「爆発」という感じでイメージしてください。この爆発オブジェクトは，弾に当たっても死にませんし，プレイヤー機に当たってもダメージを与えません。つまり，当たり判定が皆無のオブジェクトとして定義します。また，一定時間その場で表示されると，自動的に自分を削除するように移動処理を組んでください。

できました？　ここまでくればおわかりですね。この爆発オブジェクトを，敵機が弾に当たり，画面から消えていく前に，その敵機とまったく同じ場所に作成していけばいいのです。これで簡単解決，爆発の演出を行うことができました。また，弾から出る煙など，細かな演出も同様の方法でできることを思いつくのはさほどむずかしいことではないでしょう。

それだけじゃありません。画面全体に対する演出も，これとほとんど同様の方法で行うことができます。今回お話したシナリオ・データは，時間軸に合わせて自由にオブジェクトを生成することができるようになっています。そして，オブジェクトのサイズは，何も敵機や弾といった小さなサイズだけではありません。画面全体を覆う雲のようなオブジェクトを作ったっていいわけです。当たり判定さえ無視してしまえば，どんなオブジェクトだって画面演出に花を添えるものになってしまいます。あ，もちろん，動作環境との処理速度に関する相談は必要ですけどね。

それでもダメなときはシナリオ・データと同様に，演出用シナリオ・データを作り出す他ありません。たとえば，今回作ったSCENARIOOBJECT構造体のメンバ，nCharaTypeはint型の表現力を持つ変数です。int型という変数は，マイナスの値も扱うことができますが，通常，機体の種類をマイナスで表現することはしないでしょう。人間，何かに番号を振るときは，正の整数を使うものです。というわけで，おそらくマイナスという数字は使われないで余っていることでしょう。そこで，この部分にマイナスの値が入ってきた場合は，特例演出番号ということで，オブジェクトだけではどうにもならない演出を起動させるキーにしてしまいましょう。

これを強引なやりかただと思わないでください。データの数値に，二つ以上の意味を持たせるというのはごく当たり前のように行われることです。とくにメモリが少ない携帯ゲーム機や，昔のパソコンでは当然のように使われたテクニックです。もちろん，メモリが大量にある環境なら，演出用に変数を追加してもいいかもしれませんし，おそらくそちらのほうがプログラムもすっきりします。が，こういう方法があるということも知っておいて損はないと思います。

最後に

というところで，メインループから見たシューティング・ゲームのお話，いかがだったでしょうか。シューティング・ゲームに限らず，時間軸をメインに動く場合のシナリオ管理，そして画面に出る多くのオブジェクトの管理方法について，参考になればうれしいです。

さて次回は，シューティング・ゲーム編のまとめとして，よくゲームで見かけるアルゴリズムを紹介したいと思います。そうです，誘導弾とか…お楽しみに！

catlike最新情報

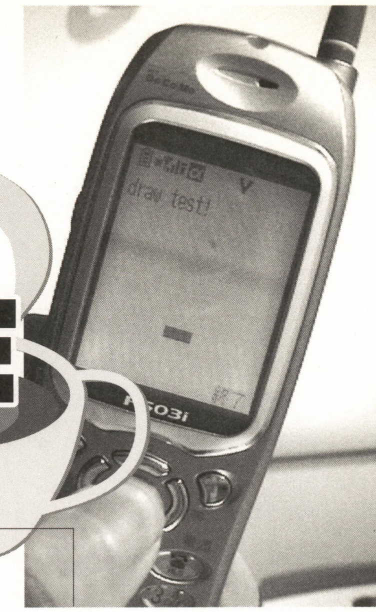
なんか，トラブル続いて，予定では10月号が発売するころにはWeb公開できていたはずの『いつか，何処かで。』ですが，この原稿を書いている現在，できてません。ごめんなさいー。あとほんのわずか，というところなので，この11月号が出るころには，きっと，多分…というところ。理由はWebページにも掲載したので，駄文と思いつつ見てくださいませ。

そんなわけで今回もスナップショットは第1話の1シーン。主人公のヒロちゃん，制服のお姉さんと待ち合わせして，これからどこへ行くのかな…？



携帯電話で動くゲームを作ろう!!

君にもできる「iアプリ」制作



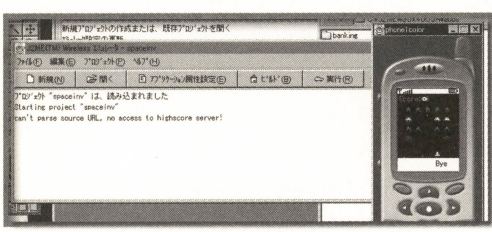
第7弾「タイマを使ったｉアプリに挑戦」

NTTドコモの発表では，ｉアプリ（503i）利用者数はなんと645万人だそうです（9月16日現在）。FOMAのサービス開始でますます活気づくｉアプリです（松原拓也）。

ｉアプリの作りかた

①サンマイクロシステムズのWEBサイト（http://java.sun.com/）からJava2の開発環境「Java 2 SDK Standard Edition 1.3.0」をダウンロードします。インストールすると“C:¥jdk1.3”というディレクトリにJavaコンパイラが展開されます。

② NTTドコモのWEBサイト（http://www.nttdocomo.co.jp/i/java.html）からｉアプリの開発ツール「J2ME Wireless SDK for the DoJa release2.2（以下，DoJa）」をダウンロードします。ダウンロードには名前の登録が必要です。インストールすると“C:¥J2MEWSDK4DOJA”フォルダに主要なファイルが展開されます。



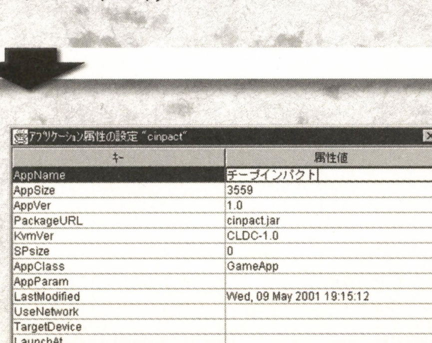
③スタートメニューから「J2ME～DoJa」を選択して，「KToolBar」を起動します。

④「新規」ボタンを押して，新規プロジェクト名を入力すると，“C:¥J2MEWSDK4DOJA¥apps”フォルダの下にプロジェクト名のフォルダと，その下に“bin”，“res”，“src”フォルダが作成されます。

⑤“res”フォルダに画像（gif）や音楽ファイル（mld），“src”フォルダにはソースファイル（java）を格納します。

⑥「アプリケーション属性設定」ボタンを押して，キーの値を入力します（“AppSize”と“Last Modified”は自動的に更新されます）。

AppName =（アプリケーション名）
AppVer = 1.0
AppClass =（最初に実行するクラス名）
SPsize =（スクラッチパッドのサイズ）
KvmVer = CLDC-1.0

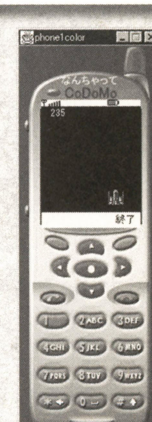


アプリケーション属性の設定 "cinpact"

キー	属性値
AppName	チープインパクト
AppSize	3559
AppVer	1.0
PackageURL	cinpact.jar
KvmVer	CLDC-1.0
SPsize	0
AppClass	GameApp
AppParam	
LastModified	Wed, 09 May 2001 19:15:12
UseNetwork	
TargetDevice	
LaunchAt	

⑦「ビルド」ボタンを押すと，コンパイルとプリベリファイが一括して行われ，jarファイルとjamファイルが“bin”フォルダに生成されます。jarファイルは"res"フォルダの中身とクラスファイルがzip形式で圧縮されたものです。

⑧エディタで“C:¥J2MEWSDK4DOJA¥lib¥devices¥phone1 color”フォルダの“phone1 color.properties”を以下のように書き換えます（値は120×130ピクセル時の設定です）。書き換えたら「編集－設定」をメニュー選択して，「デバイス指定」を“phone1 color”にします。




```
 :
screen.x=27
screen.y=65
 :
screen.width=120
screen.height=156
 :
icon.up: 44, 138, off
 :
icon.down: 44, 146, off
 :
screenPaintableRegion.width=120
screenPaintableRegion.height=130
 :
softbutton.0=0,138,36,18, softButton
softbutton.1=90,138,36,18, softButton
 :
```

⑨「実行」ボタンを押すと，携帯電話エミュレータ上でｉアプリが実行されます。

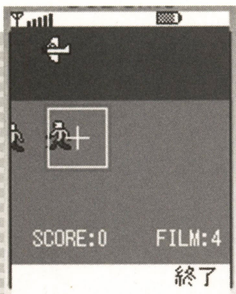
タイマを使ったｉアプリに挑戦

今回はタイマ機能のクラスを使ったゲームを作ってみました。「タイマ」は過去に紹介した「スレッド（Thread）」と性能的に同じですが，スレッドが独立したタスクなのに対して，タイマは一定時間おきにイベントが発生するという動作上のちがいがあります。

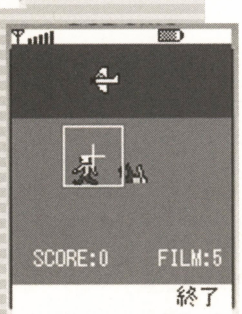
『写真ボーイ』

かのPCエンジン用のゲームを参考に作ってみました。方向キーを使ってファインダー（四角いワク）を動かし，選択ボタンで被写体を撮影してください（**写真1**）。

被写体は「飛行機」「普通の人」「背広の人」の3種類です。それらは，UFOに



◀《写真1》狙いを定めて写真を撮るゲーム



▶《写真2》人が転んだ。シャッター・チャンス！（……なんか不謹慎）



◀《写真3》UFOだ！と思ったら，時間切れでゲーム・オーバー

《リスト1》HTML ファイル "photo.html"

```
<HTML>
<HEAD>
<TITLE>i-appli</TITLE>
</HEAD>
<BODY>
ｉアプリゲーム<BR>
<OBJECT declare id="b" data="photo.jam" type="application/x-jam">
RPG</OBJECT><BR>
<A ijam="#b" href="rpg.html">写真ボーイ</A>
<br>
</BODY>
</HTML>
```

《リスト2》JAM ファイル "photo.jam"

```
PackageURL = photo.jar
AppSize = 3959
AppName = 写真ボーイ
AppVer =
AppParam =
AppClass = GameApp
SPsize = 0
LastModified = Tue, 11 Sep 2001 16:23:38
KvmVer = CLDC-1.0
```

《リスト3》Java ファイル "GameApp.java"

```
/*------------------------------------------------------------*/
/* "GameApp.java"  by 松原拓也*/

import com.nttdocomo.util.*;  /*クラスのインポート*/
import com.nttdocomo.ui.*;

public class GameApp extends IApplication
{
    final int SPEED = 100;     /* ゲームスピード*/
    final int TIMERID = 0;     /* タイマID*/

    public void start()
    {
        GameScreen screen = new GameScreen(); /*キャンバスの作成*/
        Display.setCurrent(screen);  /*キャンバスのカレント表示*/

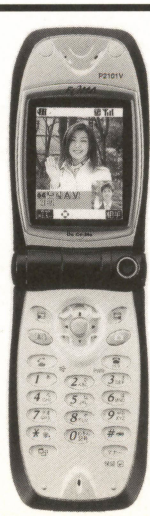
        ShortTimer stTimer = ShortTimer.getShortTimer(screen, TIMERID, SPEED, true); ◀……❶
        stTimer.start();
    }
}
```

FOMA始動!!

http://foma.nttdocomo.co.jp



◀FOMA N2001…通話・iモード・ツールなど同時使用可能なスタンダードタイプ（シルバーとレッドの2色）



◀FOMA P2101V…カメラ機能を搭載し，カラー大画面を実現したビジュアルタイプ



◀FOMA P2401…PCカード一体型のデータタイプ

試験段階だったFOMA（フォーマ）が遂に10月からサービスを開始しました。まだエリアが東京都心周辺のみですが，来年には全国規模に拡大するようです。

FOMAの特徴はクリアな音声通話と，高速な通信機能。iモード時はパケット通信（下り最大384kb/s，上り最大64kb/s）。TV電話では64Kデータ通信（上り・下りとも64kb/s）が適用されます。

FOMA対応機種は「スタンダードタイプ（FOMA N2001）」「ビジュアルタイプ（FOMA P2101V）」「データタイプ（FOMA P2401）」の3タイプです。

変身したり，転んだり，ヅラが取れたりしますので，その瞬間を撮影すると得点になります（**写真2**）。フィルムがなくなるか，持ち時間がなくなるとゲーム・オーバーです（**写真3**）。

入力方法

リスト1～4をテキスト・エディタなどを使って入力して，さらに，“0_0.gif”～“2_1.gif”までの6枚のGIF画像をグラフィック・エディタなどで作成します（**写真4**）。あとは「iアプリの作成方法」を参考にしてビルドしてください。

技術的な説明

❶“ShortTimer.getShortTimer”でタイマ・オブジェクトを作成します。引数にはタイマのID，タイマイベントの発生間隔（ミリSec単位）などを与えます。作成したタイマは，“start”メソッドで動作を開始させます。

❷イメージ（Image）を2次元の配列で管理してみました。
1番目の添え字がキャラクターの番号，そして2番目の添え字がアニメ・パターンの状態です。たとえば，[0][0]が飛行機の絵，[0][1]がUFOの絵となります。

来年春，現行iモードが28.8kbpsに対応

http://www.nttdocomo.co.jp/

現行方式（PDC-P）のパケット通信速度は9.6Kbpsですが，来年春発売予定の端末から3倍の28.8Kbpsに対応するそうです。これに合わせてiアプリのサイズも上限が10Kバイトから30Kバイトに増える予定ですが，恩恵を受けられるのは来年春以降発売の一部の機種のみのようです。

《リスト4》Java ファイル“GameScreen.java”

```
/*---------------------------------------------------------------*/
/* 「写真ボーイ」キャンバス "GameScreen.java"  by 松原拓也 */

/* クラスをインポート*/
import com.nttdocomo.ui.*;
import com.nttdocomo.io.*;
import javax.microedition.io.*;
import java.util.*;
import java.io.*;

public class GameScreen extends Canvas {

    final int MENW = 120;   /* 画面の幅・高さ[ピクセル] */
    final int SCOPEW = 16;   /* キャラの幅・高さ[ピクセル] */
    final int CROSSW = 6;    /* ＋の幅・高さ[ピクセル] */
    final int PATMAX = 2;    /* アニメパターン数*/
    final int OBJMAX = 3;    /* 被写体の数、被写体の種類*/
    final int SEQ_START=0;        /*シーケンス：スタート*/
    final int SEQ_READY=40;       /*シーケンス：READY表示*/
    final int SEQ_GO=45;          /*シーケンス：GO表示*/
    final int SEQ_GAMEOVER=1400; /*シーケンス：ゲームオーバー*/
    final int SEQ_END=1450;       /*シーケンス：終了*/

    Image imgStar[][] = new Image[OBJMAX][PATMAX];     /* イメージ配列 */
    int iTx[] = new int [OBJMAX];  /*被写体の座標*/
    int iTy[] = new int [OBJMAX];
    int iTp[] = new int [OBJMAX];
    int iTb[] = new int [OBJMAX];
    int iX ,iY ,iShoot;    /*ファインダーの座標、撮影用カウンタ*/
    int iSx,iSy,iFilm,iScore;  /*撮影時の座標、フィルム数、スコア*/
    int iSeq;     /*シーケンス*/
    Random rnd = new Random();   /* 新しい乱数列を作成*/

    /*-------新しい被写体の登場*/
    public void  objBone(int i) {
        iTp[i] = Math.abs(rnd.nextInt() % OBJMAX);
        iTx[i] = MENW;
        iTy[i] = (iTp[i]*(MENW/3)) + (SCOPEW/2)
                + Math.abs(rnd.nextInt() % ((MENW/3)-SCOPEW));
        iTb[i] = 0;
    }

    /*-------プレイデータのクリア*/
    public void objInit() {
        int i;
        for(i=0 ;i < OBJMAX ;i++) {
            objBone(i);
        }
        iSeq = SEQ_START;
        iX = MENW/2;
        iY = MENW/2;
        iFilm = 10;
        iScore = 0;
        iShoot = SCOPEW;
    }

    public GameScreen() {    /*コンストラクタ*/
        int i,j;
        MediaImage mi;

        setSoftLabel(Frame.SOFT_KEY_2, "終了");

        /*------GIF画像の一括ロード*/
        for(i=0 ;i < OBJMAX ;i++) {
            for(j=0 ;j < PATMAX ;j++) {
                    mi = MediaManager.getImage("resource:///"+ i+"_"+j +".gif");
                    try {
                            mi.use(); /* イメージの読み込み */
                    } catch (ConnectionException e) {
                    }
                    imgStar[i][j] = mi.getImage(); /* GIF画像読込 */   ◀………………… ❷
            }
        }
        objInit();
    }

    public void paint(Graphics g) {  /*ペイントメソッド*/
        int i;

        g.lock();
        g.setColor(g.getColorOfRGB(0, 0, 255));  /*空*/
        g.fillRect(0, 0, MENW ,MENW/3  );
        g.setColor(g.getColorOfRGB(128, 128, 128));  /*地面*/
        g.fillRect(0 ,MENW/3 ,MENW ,MENW );

        for(i=0 ;i<OBJMAX ;i++) {
            g.drawImage(imgStar[iTp[i]][iTb[i]] ,iTx[i]-(SCOPEW/2) ,iTy[i]-(SCOPEW/2) );
        }

        g.setColor(g.getColorOfRGB(255, 255, 255)); /*ファインダーの表示*/
        g.drawLine(iX        ,iY-CROSSW ,iX        ,iY+CROSSW);
```

```
        g.drawLine(iX-CROSSW ,iY        ,iX+CROSSW ,iY   );
        g.drawRect(iX-SCOPEW ,iY-SCOPEW ,SCOPEW*2  ,SCOPEW*2 );

        if(iShoot > SCOPEW) {  /*撮影時の赤ワク表示*/
            g.setColor(g.getColorOfRGB(255, 0, 0));
            g.drawRect(iSx-iShoot ,iSy-iShoot ,iShoot*2 ,iShoot*2 );
        }

        g.setColor(g.getColorOfRGB(255, 255, 255));
        g.drawString("SCORE:"+iScore+" FILM:"+iFilm, 12, 120);

        if(iSeq < SEQ_READY) {  /*テロップの表示*/  ◀···❸
            g.drawString(" READY..." ,30 ,20 );
        }else if(iSeq < SEQ_GO){
            g.drawString("  ..GO!! " ,30 ,20 );
        }else if(iSeq < SEQ_GAMEOVER){
            /**/
        }else if(iSeq < SEQ_END){
            g.drawString("GAME OVER" ,30 ,20 );
        }
        g.unlock(false);  /* ロック解除 */
    }

    /*---------イベント用メソッド*/
    public void processEvent(int nType, int nParm)
    {
        int i;
        int iKey;

        if (nType == Display.TIMER_EXPIRED_EVENT) {  ◀···❹
           iKey=getKeypadState();        /* 主人公の移動 */
           if(iKey == (1 << Display.KEY_SOFT2))
               IApplication.getCurrentApp().terminate();

           for(i=0 ;i<OBJMAX ;i++) {  /*被写体の移動処理*/  ◀···❺
               iTx[i] -= Math.abs(rnd.nextInt() % 3);
               if(iTx[i] < 0) {
                   objBone(i);  /*画面外に出た場合の処理*/
               }
               if(Math.abs(rnd.nextInt() % 80) < 1)
                   iTb[i] = iTb[i] ^ 1;  /*被写体の変化*/
           }

           iSeq++;
           if(iSeq < SEQ_GAMEOVER) {
               /* ファインダーの移動 */
               if(iKey==(1 << Display.KEY_LEFT))
                   iX-=8;

               if(iKey==(1 << Display.KEY_RIGHT))
                   iX+=8;

               if(iKey==(1 << Display.KEY_UP))
                   iY-=8;

               if(iKey==(1 << Display.KEY_DOWN))
                   iY+=8;

               if(iShoot <= SCOPEW) { /*撮影時の処理*/  ◀···❻
                    if(iSeq > SEQ_GO) {
                        if(iKey==(1 << Display.KEY_SELECT)) {
                            iShoot=SCOPEW+5; /*撮影開始*/
                            iSx=iX;
                            iSy=iY;
                        }
                    }
               }else{
                    iShoot--;
                    if(iShoot <= SCOPEW) { /*当り判定*/
                        for(i=0 ;i<OBJMAX ;i++) {
                            if(Math.abs(iTx[i] - iX)<SCOPEW) {
                                if(Math.abs(iTy[i] - iY)<SCOPEW) {
                                    if(iTb[i]==1)
                                        iScore+=10; /*撮影成功*/
                                }
                            }
                        }
                        if(--iFilm <= 0)
                            iSeq=SEQ_GAMEOVER;  /*フィルムが無くなった*/
                     }
               }
           }else{
               if(iSeq > SEQ_END) {
                   objInit();  /*ゲームオーバー表示完了*/
               }
           }
           repaint(); // 再描画する。
       }
    }
}
```

❸いきなりゲームが始まってしまうと，節操がないので「READY...」「GO!」というメッセージが順番に出るように演出しました。文字の移り変わりはシーケンス用の変数で管理しています。

❹イベントのタイプが“TIMER_EXPIRED_EVENT”の場合，タイマによるイベントです。このプログラムでは100ミリSecおきに呼び出されるよう設定しています。

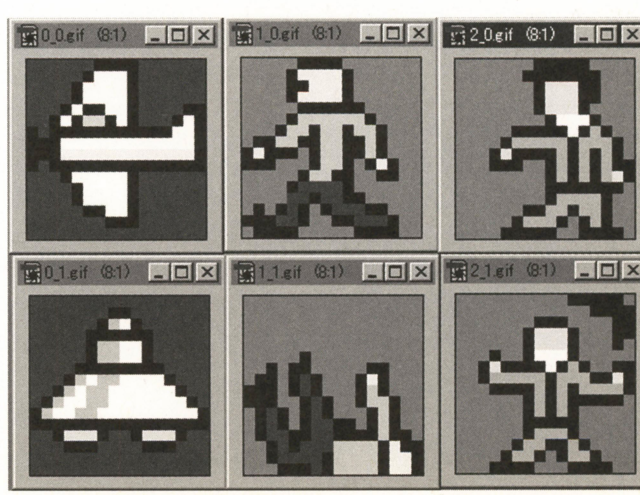
❺見た目の工夫ですが，ゲーム・オーバーになっても被写体が動き続けるようにしています。それと，被写体が転んだりするタイミングは乱数で判定しています。パターン・ゲームにしたい場合，配列を参照するように改造するといいでしょう。

❻選択（ｉ）ボタンを押してから，被写体の当たり判定までに時間差を設けました。こうすることで，予想して動かさないといけなくなるので，ゲーム性が少し上がったのではないでしょうか。

最後に

今月はいかがだったでしょうか。ｉアプリ用の投稿プログラムも月を追うごとに充実してきました。

この連載内容を参考にして，おもしろいｉアプリの投稿を期待しています。



▲《写真4》画像は左から順（上・下）に“0_0.gif”“0_1.gif”“1_0.gif”“1_1.gif”“2_0.gif”“2_1.gif”の６枚

松原拓也の自作一代記（じさくいちだいき）　連載8回目

その①ワンダースワンにキーボードを接続

「PS/2キーボード・プロトコル・コンバージョン・アダプタ」

今月は，一気に二つの製作をやってみます。（松原拓也）

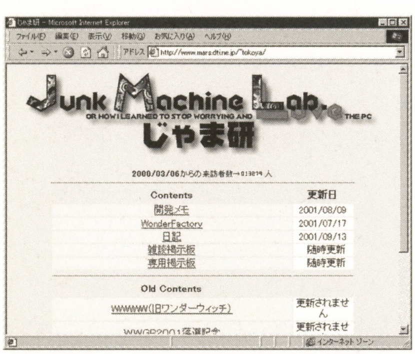


ワンダーウィッチ・ユーザーのtokoyaさんが，「PS/2キーボード・プロトコル・コンバージョン・アダプタ(以下，PPCA)」をインターネット上で公開しています（画面1，2）。これはワンダースワンにキーボードを接続してしまうという自作のハードウエアです（第1図）。

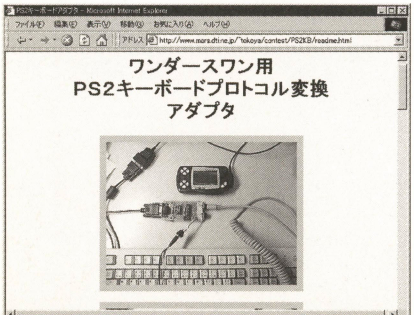
キーボードを制御する技術に興味をそそられたので，このPPCAを製作してみたいと思います。

部品の購入

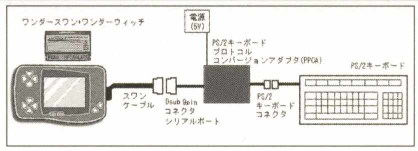
まずは以下の部品を用意します。主要な部品はワンチップマイコンと通信用のICだけとシンプルです。



▲《画面1》tokoyaさんのWEBサイト「じゃま研」
http://www.mars.dti.ne.jp/˜tokoya/



▲《画面2》「PS/2キーボード・プロトコル・コンバージョン・アダプタ」の紹介ページ



▲《第1図》概念図。PS/2キーボードを自作ハードで仲介してワンダースワンに接続する

・AT90S2313-10（370円）

ATMEL社のAVRというシリーズのワンチップマイコンです。8bitのプロセッサとRAM，フラッシュROM，EEPROM，そして15個のポートを内蔵しています（写真1）。

・ADM232A（計280円）

通信用のドライバICです。CMOS/TTL用の電圧（3～5V）をRS232用（－10～10V）に変換します。0.1μFのセラミックコンデンサ4個が必要です（写真2）。

・セラミック発振子（40円）

AVRにクロックを与える部品です。周波数は8MHzです。

・LM7805（計110円）

5V出力の3端子レギュレータです。LM2940でもかまいません。0.1μFのセラミックコンデンサと10～100μF程度の電解コンデンサが必要です。

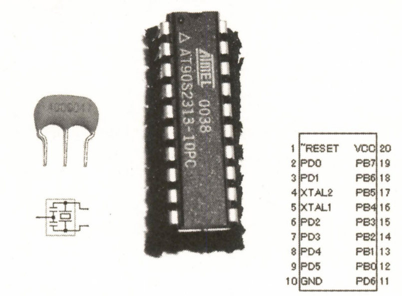
・DIN6pinコネクタ（200円）

メス型です。今回は余ったキーボード延長ケーブルを切断して流用しました。

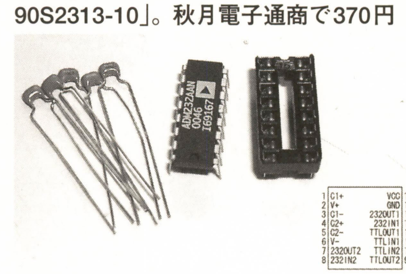
・ユニバーサル基板（70円）

・Dsub9pinコネクタ（50円）

パソコンのシリアルポートに直接接続する場合，メス型にします。



▲《写真1》ワンチップマイコン「AT90S2313-10」。秋月電子通商で370円



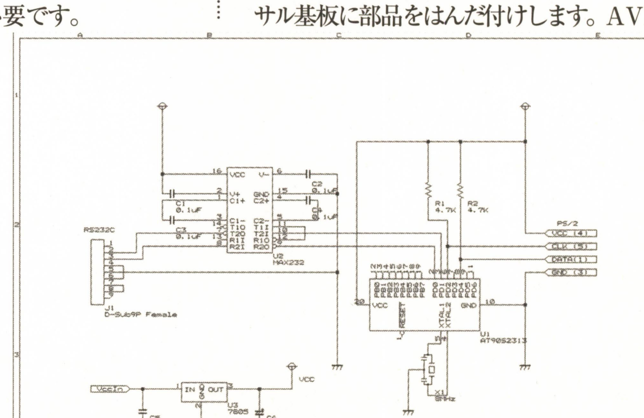
▲《写真2》通信用IC「ADM232A」。秋月電子通商で280円

・キーボード

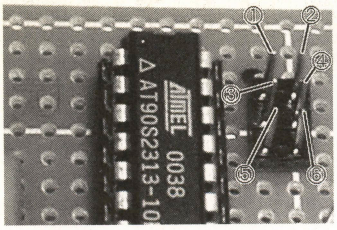
PS/2規格のキーボードです。

組み立て

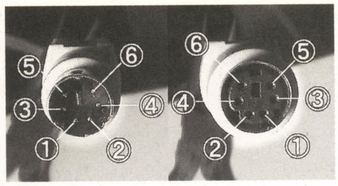
回路図（第2図）を見ながらユニバーサル基板に部品をはんだ付けします。AVR

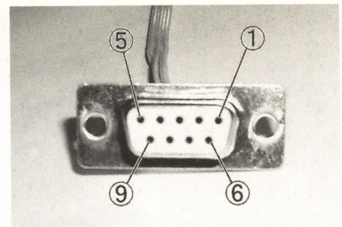


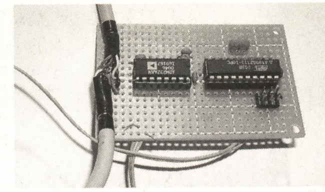
▲《第2図》PPCAの回路図（tokoyaさん作）

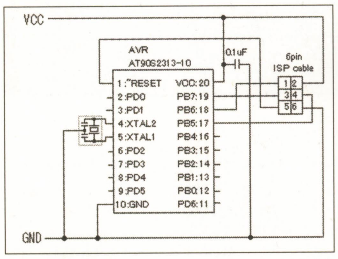


▲《写真3》ISPコネクタのピン配置

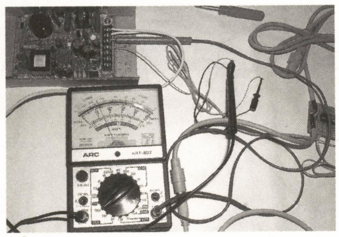

▲《写真4》DIN6pinオス（左），メス（右）のピン配置

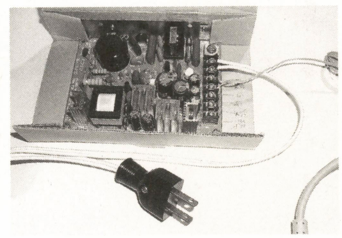

▲《写真5》Dsub9pinコネクタのピン配置

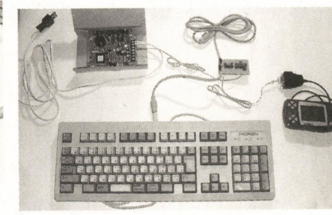

▲《写真6》ハードウエアの完成



▲《第3図》ISPコネクタの配線方法


▲《写真7》テスタを電流計に設定して，回路の消費電流を測定


▲《写真8》5V出力可能なスイッチング電源


▲《写真9》完成したPPCA

にはVCCとGND，クロック，ISPコネクタ（**写真3**）を取りつけます（**第3図**）。キーボードコネクタ（**写真4**）と通信用IC，Dsub9pin（**写真5**）をつければ完成です（**写真6**）。

あとはVCCとGNDを電源につなぎます。この回路をテスタで測定してみたら，合計150mAほど電流を消費していました（**写真7**）。そこで，それ以上の出力を持った9VのACアダプタか，5Vのスイッチング電源（**写真8**）を接続します（この場合，3端子レギュレータが不要になります）。ハードウエア完成です（**写真9**）。

プログラミング

tokoyaさんのWEBサイトからAVRに書き込むファームウエア用のソース（“CodeTbl.inc”“PPCA.asm”）をダウンロードします。

アセンブラはATMEL社からダウンロードします（無料）。ヘッダファイル“2313def.inc”も含まれています。アセンブラを使ってソースをアセンブルしてHEXファイルを生成させます（**画面3**）。HEXファイルはISPコネクタにAVRライタを接続してから，電源を入れて書き込みます（**写真10**）。

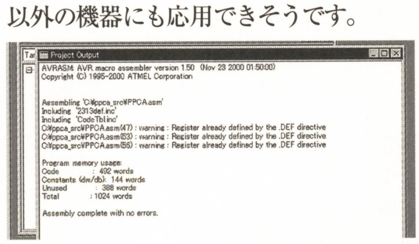
ワンダースワンとの接続

動作の検証にワンダースワン側のプログラムが必要です。そこで，1バイトずつ受信して画面に表示する簡単なソフトを自作しました。通信設定は38,400bps（データビット8，ストップビット1，パリティなし）です。

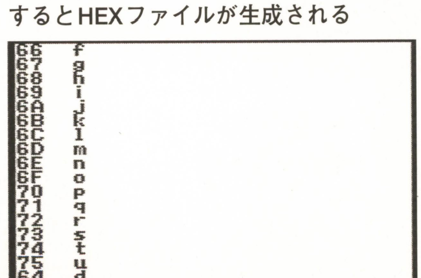
実行後，キーボードをタイプすると文字が現れました。成功です（**画面4**）。

最後に

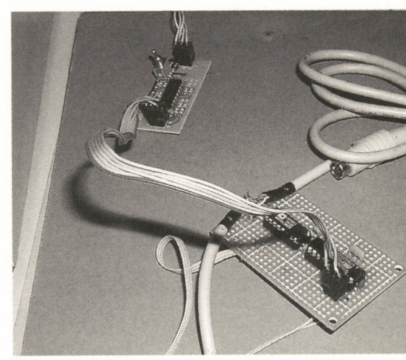
あとは使い道ですね。ワンダースワンとキーボードを持ち歩いて原稿が書けたらカッコいい（？）と思いますが，漢字変換可能なFEPがありません。電源も持ち運べるようにしないといけません。実用まであと1歩でしょう。このPPCAは出力がRS232Cなので，ワンダースワン以外の機器にも応用できそうです。



▲《画面3》ソースのアセンブルに成功するとHEXファイルが生成される



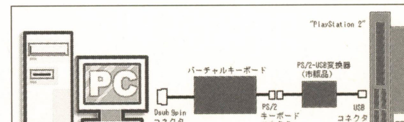
▲《画面4》自作の通信ソフトでキーボード入力中（ワンダーウィッチの画面です）


▲《写真10》シリアルポートを通じて，HEXファイルをAVRのフラッシュROMに書き込み

あと，一つアイデアが……このPPCAをまるっきり逆にして「キーボードに化ける」機能を作れないでしょうか。うまくいけばプレイステーション2用の「BASIC STUDIO パワフルゲーム工房」でプログラムの自動入力ができるのではないでしょうか（**第4図**）。DIN6pinのオス／メスを逆にして，ファームを大幅に書き換えないといけませんが。

とにかくこのPPCAの可能性に期待です。製作期間は2日。製作費用は1,120円でした。

★ ★ ★



▲《第4図》アイデアその1。こういう「バーチャルキーボード」を作れないでしょうか？

その②手ごろで入門者向けワンチップマイコン

「PICプログラマーキット Ver.3」

価格：6,700円
対応マイコン：PIC12C509, PIC16F877, PIC16C74, PIC16F84, PIC16C711など。
秋月電子通商　http://www.akizuki.ne.jp/



ワンチップマイコンとは，CPU，RAM，ROM，I/Oを一つのパッケージにしたマイコンのことです。ワンチップマイコンは扱いが簡単で，サイズは小型，価格も安く，速度は昔のパソコン並み。電子工作の世界では広く普及しています（写真1）。

ワンチップマイコンはATMEL社の「AVR（エーブイアール）」と，MICROCHIP社の「PIC（ピック）」が有名です。どちらも8bitのRISCプロセッサを搭載したマイコンです。プログラミング環境はアセンブラです。プログラムは「ROMライタ」という機器を使ってフラッシュROMに書き込みます。

秋月電子通商の「PICプログラマーキットVer3」はPIC専用のROMライタです。通常，ROMライタは高額なものですが，これは，組み立てキットのため，手ごろで入手しやすくなっています。

キットのパーツ

キットに付属の部品は以下のとおりです（写真2）。

- ワンチップマイコン（PIC），セラミック発振子
- IC（74HC4066，SP202など）
- 抵抗，コンデンサ，LED，三端子レギュレータ
- CD-ROM（アセンブラ，ライタソフトなど）
- ゼロプレッシャーソケット（40ピン，28ピン）
- 基板

組み立て

キットに付属する説明書を読みながら，基板にはんだ付けしていきます。

部品は背の低いものから順につけていきます（写真3）。

LEDや電解コンデンサなど極性のある部品には注意しましょう（写真4）。3端子レギュレータは足を曲げてからはんだ付けします（写真5）。ICはまずICソケットを先に付け，その上にはめ込みます（写真6，7）。

電源と通信ケーブル

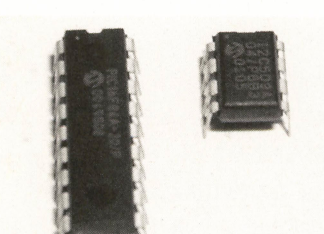
このキットは通信ケーブルとACアダプタが付属していません（「RS232Cケーブル＆電源セット」1,000円）。ACアダプタは15～24Vの電圧に対応していますので，身近な電化製品から流用してみました（写真8，9）。

RS-232Cケーブルもあり合わせのもので代用しました。通信ケーブルでパソコンとPICプログラマを接続したら完成です（写真10）。

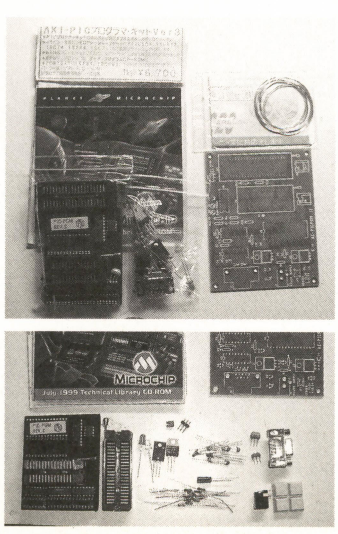
プログラム

キットには2種類のPICが付属しますので，すぐにプログラムを始めることができます。

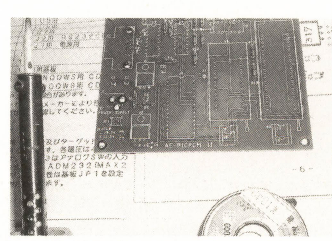
「PICプログラマ」の制御には「ライタソフト」というソフトを使います。ライタソフトはDOS用（NEC98／PC-AT



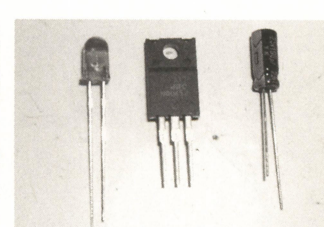
▲《写真1》付属するPIC
・「PIC16F84A-20/P」（価格：380円，FLASHROM:1Kワード，RAM:68バイト，EEPROM:64バイト，CLOCK:最大20MHz）
・「PIC12C509A-04/P」（価格：150円，ROM:ワンタイム1024×12bit，RAM:41バイト，EEPROM：なし，CLOCK:最大4MHz）



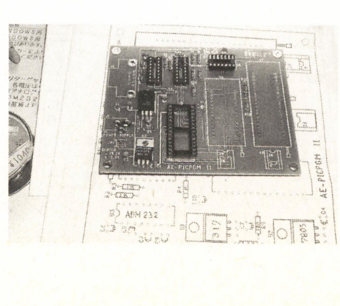
▲《写真2》キット内のパーツ



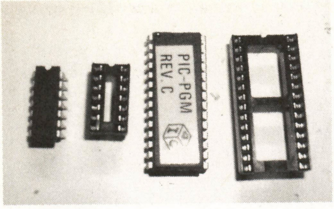
▲《写真3》背の低いパーツから取りつけ



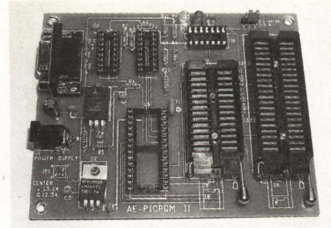
▲《写真4》LED（長いほうがアノード），3端子レギュレータ，電解コンデンサ（長いほうが＋）の極性に注意



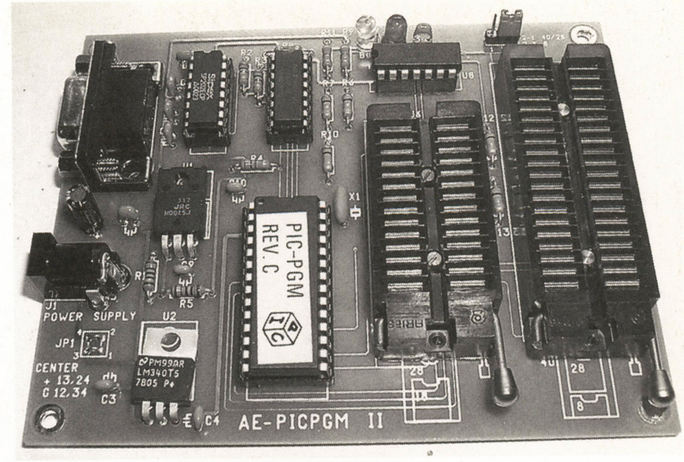
◀《写真5》3端子レギュレータは足を90度曲げてからはんだ付け



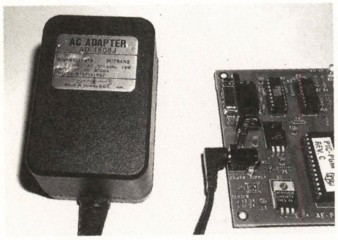
▲《写真6》ICとICソケット，制御用のマイコン（PIC）



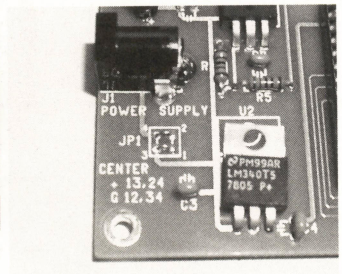
▲《写真7》はんだ付けしたICソケットの上に装着する



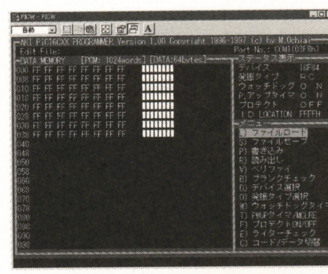
▲《写真10》完成した「PICプログラマーキット」



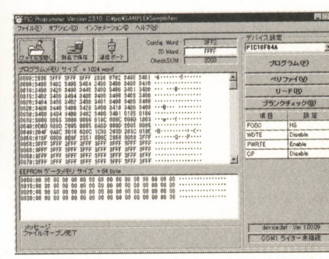
▲《写真8》ACアダプタ15-24V



▲《写真9》ACアダプタの電極に合わせてジャンパを設定



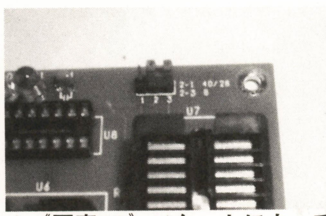
▲《写真13》DOS用ライタソフト“Picw.exe”



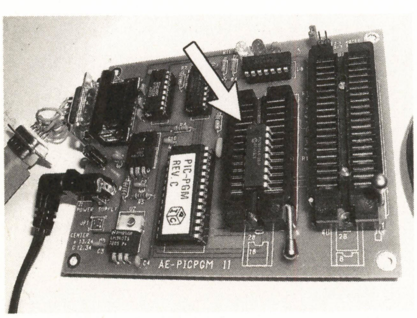
▲《写真14》Windows用ライタソフト「PIC Programmer」



▲《写真11》付属のCD-ROMにはアセンブラ，ライタソフト，サンプルプログラムを収録



▲《写真12》ソケットによってディップスイッチを設定



▲《写真15》ゼロプレッシャーソケットにPICを装填してプログラム

両対応）とWindows用の2種類があり，付属のCD-ROMに収録されています（写真11）。書き込み手順は以下のとおりです。

①まず，ソースを“PA.EXE”でアセンブルしてHEXファイルを作ります。

②PICプログラマのソケットを設定して電源を入れます（写真12）。

③ライタソフトを起動して，PICの種類を選んでHEXファイルを読み込みます（写真13，14）。

④ゼロプレッシャーソケットにPICを装てんします（写真15）。

⑤「プログラム」をメニュー選択するとプログラムが書き込まれます。

フラッシュROMは最低1000回くらい書き換えが可能ですが，ワンタイムは1回書き込んだら2度と修正ができませんので注意しましょう。

最後に

マイコンが安価でしかも手軽にプログラムできるなんて，いい時代になりました。電子工作になじみのない人も，これを機に挑戦してみてはいかがでしょうか。

テクノ・トイズ・ラボラトリ
Techno Toys Laboratory

デジタルとアナログの境界線上で作る電子オモチャ

今月の製作

ポケットライト・デラックス 前編

テクノ・オモチャ研究所

TEXT&製作●クワクボリョウタ



デバイス・アーチスト。1998年頃から「ビット」をキーワードにメディアアート作品や電子オモチャなどを製作，発表。僕が最初に青色LEDを見かけたときは一つ5,800円でした。で，意を決して買いに行ったら2,900円になってた。今では300円くらいで手に入ります。www.vector-scan.com

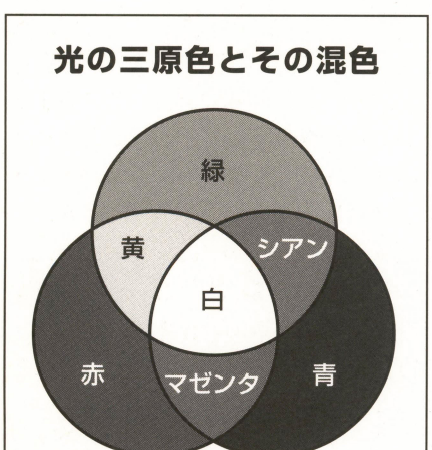
電子回路で光を作り出すのにもっとも手軽な素子は，なんと言ってもLED（発光ダイオード）だろう。最近では赤・緑・青のLEDをワンパッケージに収めたフルカラーLEDも入手できるようになった。今回はこのフルカラーLEDを使って，光の色をさまざまに調節できるポケットライト・デラックスを製作しようと思う。

いろいろな色を出すには

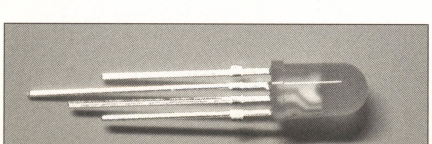
赤・緑・青。これらは光の三原色と呼ばれる。RGBという言葉をよく耳にすると思うけど，これはRed，Green，Blueの頭文字をとったものだ。

三原色を組み合わせると，あらゆる色を作り出すことができる。

たとえば，赤＋緑＝黄。緑＋青＝シアン（水色に近い色）。青＋赤＝マゼンタ（赤紫）。そして，赤＋緑＋青＝白。…といった具合に色は混合され，またその比率を変えることでさまざまな中間色が得られる（**第1図**）。フルカラーLEDは，この三原色の発光をそれぞれ独立して制御することで，あらゆる色を再現できる電子パーツだ。



《第1図》赤・緑・青の3色の組み合わせでさまざまな色が作りだせる

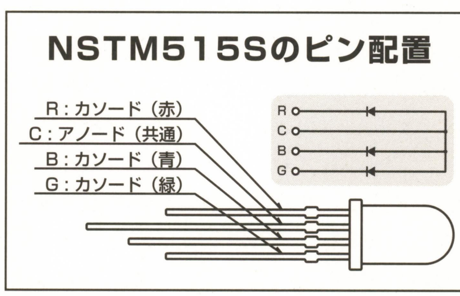


《写真1》フルカラーLED

フルカラーLEDで遊んでみよう

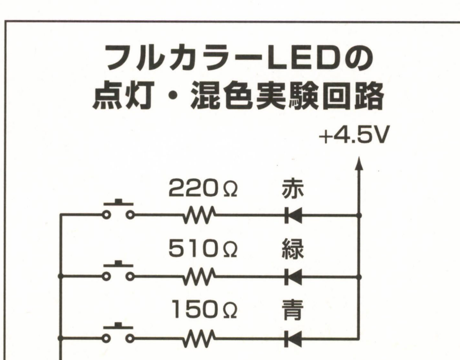
実際にフルカラーLEDを点灯して遊んでみよう。

まずはパーツ調達だ。今回は日亜化学工業のフルカラーLED，NSTM515Sを使うことにした（**写真1**）。店頭価格で2千円弱と少々高価だけど，混色の美しさは一見の価値あり！　このNSTM515Sは，赤・緑・青それぞれに発光するLEDが白濁色のプラスチックにパッケージされており，3色の光がその中で混じりあうようになっている。ピン配置は**第2図**のとおり。3色それぞれのLEDのカソード（マイナス側）と共通のアノード（プラス側）で計4本だ。



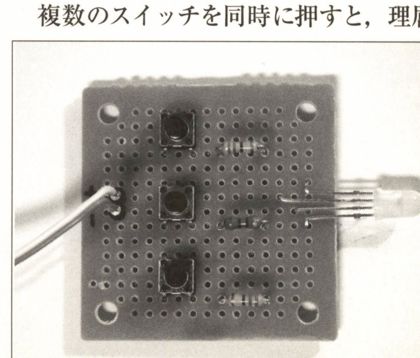
《第2図》3色のLEDがワン・パッケージになっている

混色の様子を試してみるために**第3図**，**写真2**のような実験回路を組んでみた。それぞれ抵抗を介してLEDに電流を供給している。この抵抗で各色の輝度のバランスをとり，3色同時点灯時になるべく真っ白に見えるようにしてある。抵抗値は各自の好みで調節してもかまわない。

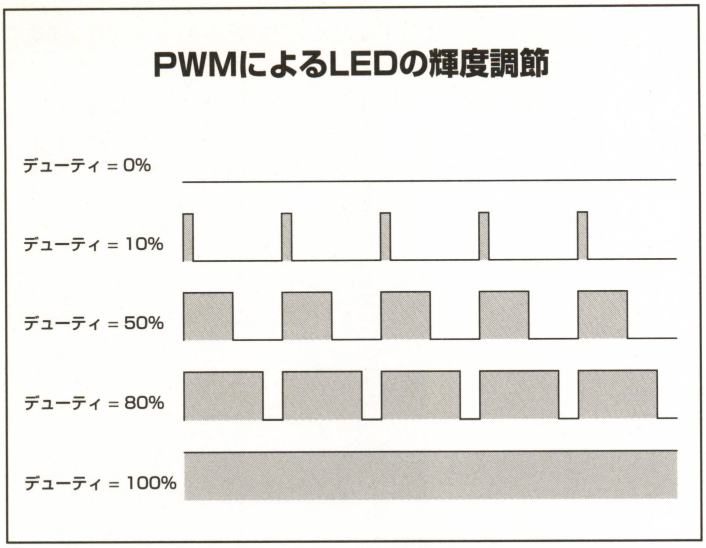


《第3図》それぞれのスイッチONの組み合わせで八つの色を作る

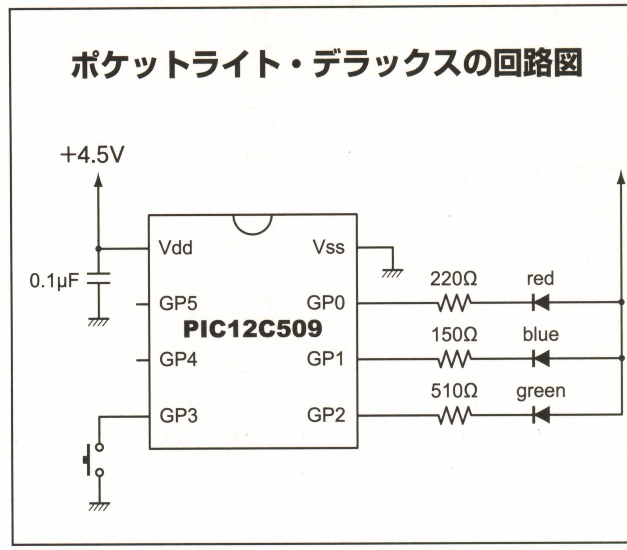
複数のスイッチを同時に押すと，理屈



《写真2》実験基板



《第4図》ONとOFFを高速でくり返すと，視覚的には中間の輝度が感じられる



《第5図》この回路図をよく見て，写真3のように配線しよう

どおり混色されるのがわかるだろう。この回路はスイッチングの組み合わせで8色（黒＝真っ暗も含む）の表示ができる。

微妙な色合いを作るには

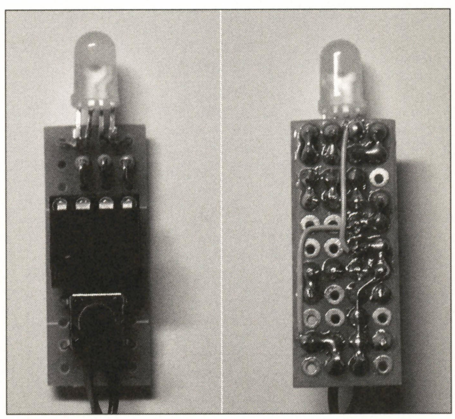
せっかくのフルカラーLED，8色発光だけじゃつまらないので，もっと多彩な色を出す方法を考えよう。そのためには3色それぞれLEDの輝度を微妙に変えることが必要になる。

PWM（Pulse Width Modulation）という方法を用いると，デジタル回路で輝度を制御することができる。その原理を**第4図**に示す。要はLEDを素早く点滅させ，発光時間の長さを変えることによって，見た目の明るさを変化させるというものだ。点滅が十分に速ければ，残像効果で見た目には点滅しているようには感じずに，輝度が変化して見える。

ここで点滅の周期のうちONになっている割合をデューティ比と呼ぶ。デューティ100%ならつきっぱなし，0%なら消えたまま，50%ならちょっと暗めに光って見える，ということになる。

PICでPWM制御

PWM制御をマイコンを使ってやってみよう。今回使うのはPIC12C509という石で，これは当研究所では初お目見えだが，今まで使ってきたPIC16F84より規模の小さい8ピンのワンチップ・マイコンだ。PIC16F84とくらべると，使える命令数が少なく，その他にもさまざまな制約もあるけれど，今回の用途にはこれで十分だし，安くコンパクトに作れる。



《写真3》ポケットライト・デラックス基板

マイコンのクロックは内部RCオシレータという機能を使い，約4MHzで動作させる。そのためクリスタルやコンデンサを外付けする必要はない。必要なパーツはマイコン本体と抵抗3本，バイパス・コンデンサ，LED，そしてタクト・スイッチ（動作をコントロールするためのもので，これは次回に詳しく説明する）。たったこれだけで実現できてしまう。**第5図**がその回路図，**写真3**が製作した基板，**第1表**がパーツ・リストとなっている。電源は乾電池3本の4.5Vを供給する。

使用するPIC12C509はライトワンス，つまり一度だけ書き込みができ，消去はできないタイプなので，予算に応じて何個か買っておいたほうがよいかもしれない。価格は一つ150円程度だ。

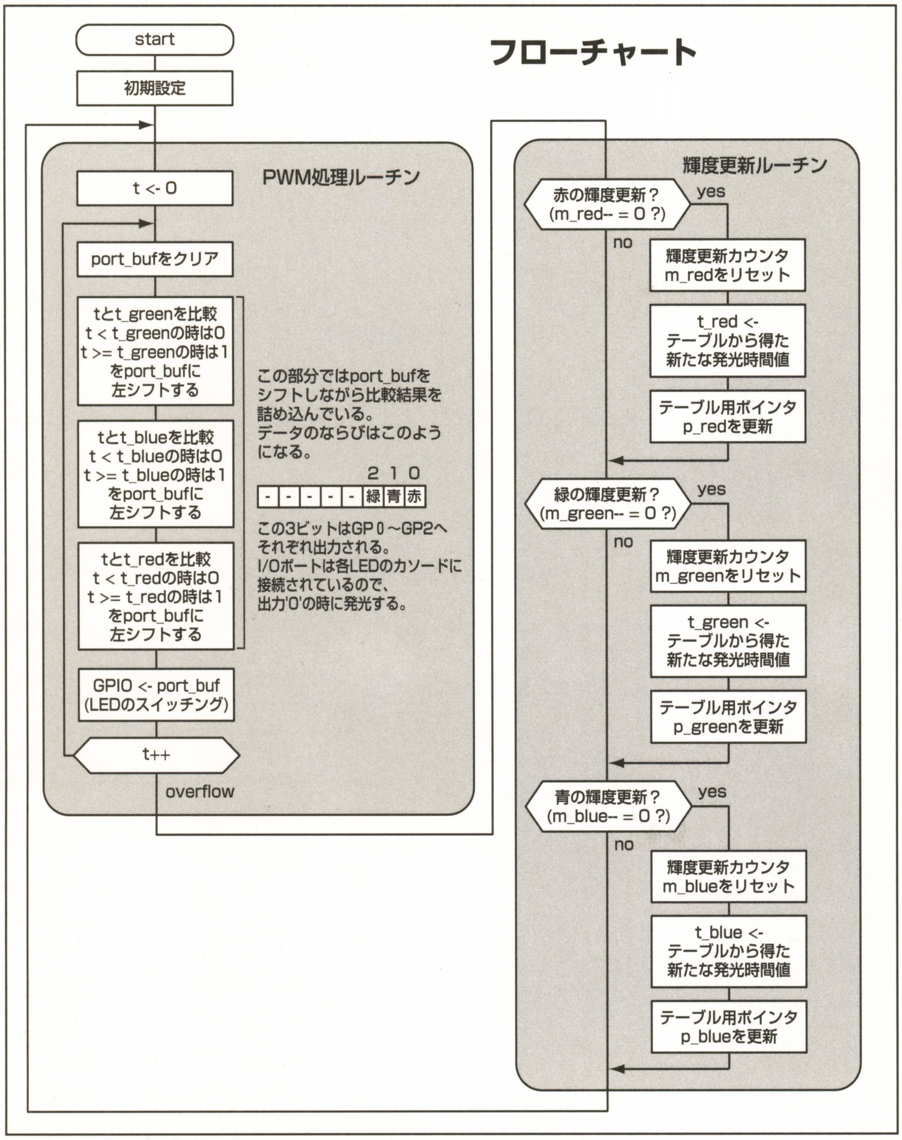
《第1表》ポケットライト・デラックスのパーツリスト

PIC12C509		x 1
ICソケット 8ピン		x 1
フルカラーLED NSTM515S		x 1
抵抗	150Ω	x 1
	220Ω	x 1
	510Ω	x 1
積層セラミックコンデンサ 0.1μF		x 1
タクト・スイッチ SOA-143HS		x 1
ユニバーサル基板		x 1
電池ボックス（4.5V。実験用なのでサイズは問わない）		

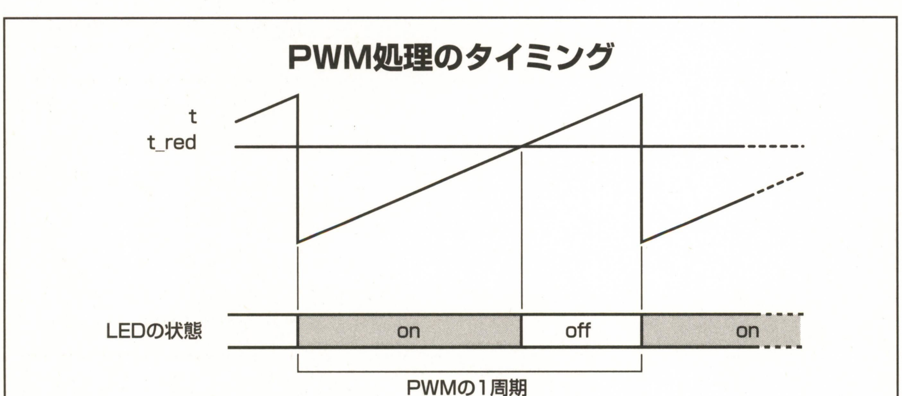
リスト1がLEDのPWM制御のテスト・プログラムだ。このプログラムは3色のLEDの輝度を徐々に変化させ続け，さまざまな色を発生する動作をする。MPASMでアセンブルし，リセットピン不使用，ウォッチドッグ・タイマ不使用，内部RCモードと指定してPICに書き込む。新品のPIC12C509には，0x3ff番地にmovlw命令が書き込まれている。これは内部RCモードで動作するときの動作クロックを4MHzに合わせるために用意されているもので，PICのリセット時にはまずこの命令が実行される。その際Wレジスタに代入される値はPICの個体ごとに違う。この番地は書き込みの際にはそのままにしておく。

プログラムの動作

プログラムについて説明しよう。**第6図**がテスト・プログラムのフローチャートだ。大きく分けて，LEDを発光させるPWM処理ルーチンと，発光色を更新する輝度更新ルーチンの二つで構成されて



《第6図》プログラムはPWM処理ルーチンと輝度更新ルーチンで構成されている



《第7図》図は赤LEDについてのもの。緑，青も同様に処理する。t_redがtよりも大きいときにLEDを点灯する。したがって，t_redが大きければ明るくなり，小さければ暗くなる

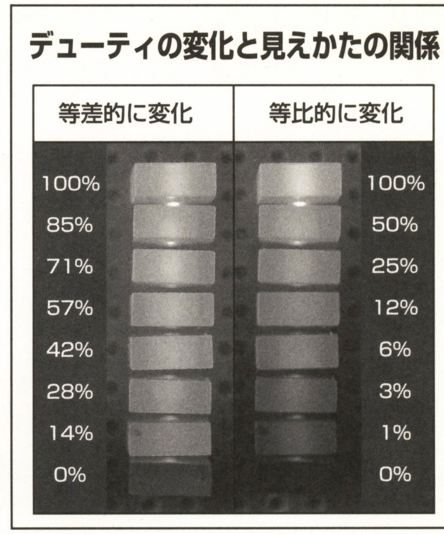
いる。

PWM処理ルーチンでは，LEDを高速に点滅させる処理を行っており，点滅の1周期を256等分してループ処理している。カウンタtが0から255までインクリメントしていき，t_red,t_green,t_blueとそれぞれ比較して，tよりそれらが大きいときだけ各LEDをONする。したがって，t_red/green/blueは各色の発光のデューティ比を決定するパラメータとして機能する。この処理のタイミングを**第7図**に示した。

一方，輝度更新ルーチンではt_red/green/blueを刻々と変化させる処理をしている。m_red/green/blueはメイン・ルーチン何周につき輝度を変更するかを決めるカウンタだ。この三つのカウンタの周期はバラバラ（ここでは19:18:17）になっているので，3色の輝度の起伏は徐々にずれていく。そうするとさまざまな比率の混色が起こるというわけだ。

t_red/green/blueの変更は，ルックアップ・テーブルから新しい値を引っぱってきて代入する。その際テーブルのインデックスとしてp_red/green/blueが使われる。p_red/green/blueはテーブル引きの後，次回のためにインクリメントされる。

テーブルには256バイトの数値が並んでいるが，これは輝度の変化を自然に見せるために等比数列になっている。**第8図**はLED点灯のデューティ比を等差的に変化させた場合と，等比的に変化させた場合を比較したものだ。一見して後者のほうが気持ちイイのが分かるでしょ？



《第8図》デューティを2とおりのやり方で変化させたグラデーション。等比的に変化させたほうが直線的な変化に見える

さて，次回は？

次回はポケットライトとして実用になるようにボディの製作をして，タクト・スイッチ一つで電源のON/OFFと色の選択ができるインターフェースのプログラミングをする。それでは！

《リスト1》ポケットライト・デラックス前編のリスト（アセンブラ）

```
;=============================================================
;
;               ポケットライト・デ ラックス
;                (テスト・プ ログ ラム)
;
;           c o d e d  b y  クワクホ゛リョウタ
;
;               PIC12C508 or 509
;                  +--------+
;         +4.5V ----|Vdd  Vss|---- GND
;              x---|GP5  GP0|---> Red cathode
;              x---|GP4  GP1|---> Blue cathode
;             SW --->|GP3  GP2|---> Green cathode
;                  +--------+
;
;=============================================================
        LIST    C=120, p=12C509, r=dec, x=off, n=0
        include <p12c509.inc>

port_buf        equu    0x07            ; LEDのｽｲｯﾁﾝｸﾞﾊﾟﾀｰﾝ
t               equ     0x08            ; 発光時間比較ｶｳﾝﾀ
t_red           equ     0x09            ; 発光時間
t_green         equ     0x0a            ;   //
t_blue          equ     0x0b            ;   //
m_red           equ     0x0c            ; 輝度更新用ｶｳﾝﾀ
m_green         equ     0x0d            ;    //
m_blue          equ     0x0e            ;    //
p_red           equ     0x0f            ; 輝度更新用ﾎﾟｲﾝﾀ
p_green         equ     0x10            ;    //
p_blue          equ     0x11            ;    //

;-------- init values ------------------
M_RED_INIT      equ     19              ;
M_GREEN_INIT    equ     18              ; 各色輝度の更新周期
M_BLUE_INIT     equ     17              ;

;=============================================================
        org     0x000

        movwf   OSCCAL                  ; 内部RCの校正

initialize:
        movlw   b'00000111'             ; I/Oの出力初期値
        movwf   GPIO                    ; (LEDをOFFにしておく)

        movlw   b'00111000'             ; I/Oの入出力設定
        tris    GPIO                    ;

        movlw   b'10011111'             ; 入力ﾋﾟﾝをﾌﾟﾙｱｯﾌﾟ
        option                          ;

        clrf    t_red                   ; 各変数の初期化
        clrf    t_green                 ;
        clrf    t_blue                  ;
        clrf    p_red                   ;
        clrf    p_green                 ;
        clrf    p_blue                  ;
        movlw   M_RED_INIT              ;
        movwf   m_red                   ;
        movlw   M_GREEN_INIT            ;
        movwf   m_green                 ;
        movlw   M_BLUE_INIT             ;
        movwf   m_blue                  ;

;=============================================================
main:
        ;=====================================================
        ;               PWM処理ルーチン
        ;=====================================================
        clrf    t                       ; tを初期化

main_10:
        ;-----------------------------------------------------
        ; tとt_red,t_blue,t_greenを比較し、
        ; それぞれがtより大きい場合に各LEDを点灯する
        ; port_buf には '00000GBR' の並びで結果が入り、
        ; 最後にまとめてGPIOに出力する
        ; (I/Oはｶｿｰﾄﾞに接続されているので0出力で点灯)
        ;-----------------------------------------------------
        clrf    port_buf                ; port_bufをｸﾘｱ

        movf    t_green,W               ; if t < t_green
        subwf   t,W                     ; then C = 0
        rlf     port_buf,F              ; port_buf
                                        ; =(port_buf<<1)+C

        movf    t_blue,W                ; if t < t_blue
        subwf   t,W                     ; then C = 0
        rlf     port_buf,F              ; port_buf
                                        ; =(port_buf<<1)+C

        movf    t_red,W                 ; if t < t_red
        subwf   t,W                     ; then C = 0
        rlf     port_buf,F              ; port_buf
                                        ; =(port_buf<<1)+C

        movf    port_buf,W              ; 結果を
        movwf   GPIO                    ;    I/Oに出力する

        incfsz  t,F                     ; tが100hになるまで
        goto    main_10                 ;          繰り返し

        ;=====================================================
        ;               輝度を更新ルーチン
        ;=====================================================
        ;-------- 赤の輝度更新 -------------------------------
red:
        decfsz  m_red,F                 ; 赤の更新ﾀｲﾐﾝｸﾞ?
        goto    green                   ;    違うならgreenへ

        movlw   M_RED_INIT              ; 更新ｶｳﾝﾀをﾘｾｯﾄ
        movwf   m_red                   ;

        movf    p_red,W                 ; t_red <-
        call    get_t                   ;    get_t(p_red++)
        bcf     STATUS,PA0              ;
        movwf   t_red                   ;
        incf    p_red,F                 ;

        ;-------- 緑の輝度更新 -------------------------------
green:
        decfsz  m_green,F               ; 緑の更新ﾀｲﾐﾝｸﾞ?
        goto    blue                    ;    違うならblueへ

        movlw   M_GREEN_INIT            ; 更新ｶｳﾝﾀをﾘｾｯﾄ
        movwf   m_green                 ;

        movf    p_green,W               ; t_green <-
        call    get_t                   ;    get_t(p_green++)
        bcf     STATUS,PA0              ;
        movwf   t_green                 ;
        incf    p_green,F               ;

        ;-------- 青の輝度更新 -------------------------------
blue:
        decfsz  m_blue,F                ; 青の更新ﾀｲﾐﾝｸﾞ?
        goto    main_90                 ;    違うなら90へ

        movlw   M_BLUE_INIT             ; 更新ｶｳﾝﾀをﾘｾｯﾄ
        movwf   m_blue                  ;

        movf    p_blue,W                ; t_blue <-
        call    get_t                   ;    get_t(p_blue++)
        bcf     STATUS,PA0              ;
        movwf   t_blue                  ;
        incf    p_blue,F                ;

main_90:
        goto    main                    ; くり返し

;=============================================================
get_t:
        bsf     STATUS,PA0              ;
        movwf   PCL                     ;

;=============================================================
;               ﾃｰﾌﾞﾙ : 発光時間の変化
;=============================================================
        org     0x200

        dt        0,   0,   0,   0,   0,   0,   0,   0
        dt        0,   1,   1,   1,   1,   1,   1,   1
        dt        1,   1,   1,   1,   1,   2,   2,   2
        dt        2,   2,   2,   2,   3,   3,   3,   3
        dt        3,   3,   4,   4,   4,   4,   4,   5
        dt        5,   5,   5,   6,   6,   6,   7,   7
        dt        7,   8,   8,   9,   9,   9,  10,  10
        dt       11,  11,  12,  12,  13,  14,  14,  15
        dt       16,  16,  17,  18,  19,  20,  21,  22
        dt       23,  24,  25,  26,  27,  28,  30,  31
        dt       32,  34,  35,  37,  39,  40,  42,  44
        dt       46,  48,  51,  53,  55,  58,  60,  63
        dt       66,  69,  72,  75,  78,  82,  86,  90
        dt       94,  98, 102, 107, 111, 116, 122, 127
        dt      133, 139, 145, 151, 158, 165, 172, 180
        dt      188, 196, 205, 214, 224, 234, 244, 255
        dt      255, 244, 234, 224, 214, 205, 196, 188
        dt      180, 172, 165, 158, 151, 145, 139, 133
        dt      127, 122, 116, 111, 107, 102,  98,  94
        dt       90,  86,  82,  78,  75,  72,  69,  66
        dt       63,  60,  58,  55,  53,  51,  48,  46
        dt       44,  42,  40,  39,  37,  35,  34,  32
        dt       31,  30,  28,  27,  26,  25,  24,  23
        dt       22,  21,  20,  19,  18,  17,  16,  16
        dt       15,  14,  14,  13,  12,  12,  11,  11
        dt       10,  10,   9,   9,   9,   8,   8,   7
        dt        7,   7,   6,   6,   6,   5,   5,   5
        dt        5,   4,   4,   4,   4,   4,   3,   3
        dt        3,   3,   3,   3,   2,   2,   2,   2
        dt        2,   2,   2,   1,   1,   1,   1,   1
        dt        1,   1,   1,   1,   1,   1,   1,   0
        dt        0,   0,   0,   0,   0,   0,   0,   0

;=============================================================
        end
```



ネット銀行・・・なんかめんどくさそう・・・とまたもや見て見ぬフリをしていたのですが，通販にハマりネットでも買い物をはじめるようになると家に居ながらにして，いろんな商品を手に入れられてこれが超便利！

ネット上に銀行口座を持つと，もっと便利になるのでは？　と，ためし口座を開いてみることにしました。

私はヤフーオークションのオフィシャルサイトであるジャパンネット銀行を選んで口座開設。マンガに描いてるとおり簡単に開設できました。早速使ってみる。なんと！　家のパソコンからアクセスするだけで，いろんな料金の振り込みが24時間可能に！　入金し忘れていた家賃を試しに振り込んでみたら，簡単にできてしまいました！次回はヤフーオークションでステキな物を落札したいと思っています。乞うご期待！

（神奈川）

ジャパンネット銀行
http://www.japannetbank.co.jp



パソコンでロボットを操る！

レゴ道場 桐林部屋



チャレンジ!! レゴ・マインドストーム

LEGO MINDSTORMS ROBOTICS INVENTION SYSTEM



お題：LEGO MINDSTORMSフェスティバル

こんにちは，桐林です。今回は8月5日に行われた第5回LEGO MINDSTORMSフェスティバルのようすです。

（文・写真／桐林星河）



桐林星河（きりばやしせいが）
機械やロボットが好きな14歳。小学生の頃からメカに慣れ親しみ，LEGO MINDSTORMSの競技などでは常に好成績。TV出演なども多数。現在中学3年の受験生。

LEGO MINDSTORMS フェスティバルとは

LEGO MINDSTORMSフェスティバルは，LEGO-MUG（レゴマグ：LEGO MINDSTORMS Users Group JAPAN）が主催しているイベントです。今回は8月5日に行われた「東京トイフェスティバル」の会場の一部を借りて行われました（**写真1～4**）。

※LEGO-MUGのホームページ……http://www.legomug.gr.jp/index.html

今回の フェスティバルの内容

今回のフェスティバルは，競技会を行わず，展示の人気投票コンテストのみでした。しかし，それでも25名38作品が出展されました（**写真5，6**）。この人気投票コンテストは，来場したお客さんの投票によって，投票数が多いほうから順位が決まっていきます。

人気投票コンテストは3部門あり，MINDSTORMSを使った作品の「MINDSTORMS部門」と，MINDSTORMSを使っていない「レゴ部門」と，作品が64ポッチ×64ポッチ以上の大きさになる「大型モデル部門」と，盛りだくさんな内容でした。

また，展示の他にも，先月号で紹介した「ツクモMUGカップ」と同じコースを使った「ミニゲーム・デモンストレーション」（**写真7**）をやったり，MINDSTORMSフェスティバル初の試みの「レゴムービー上映会」がありました。

「ミニゲーム・デモンストレーション」はその名のとおり，ツクモMUGカップのミニゲームを再現して，デモをしたものです。「レゴムービー上映会」はレゴ社から販売されているレゴスタジオシリーズを使って撮影された，レゴの短い映画などをスクリーンで上映しました。なかには海外からの出展もありました。



《写真1》展示部門付近



《写真2》プロジェクタ付近



《写真3》ミニゲーム，トレイン付近



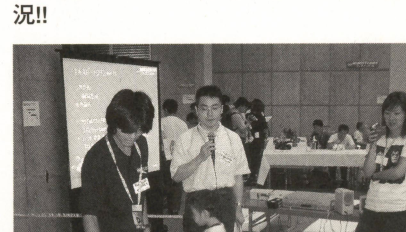
《写真4》トイフェスティバルのもよう



《写真5》トレインの展示部門。たくさんの車両が並んでいます



《写真6》展示は親子連れなどで大盛況!!



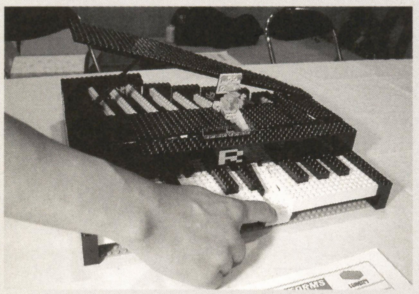
《写真7》ミニゲームのデモ風景。写真は筆者のデモ場面

レゴ部門

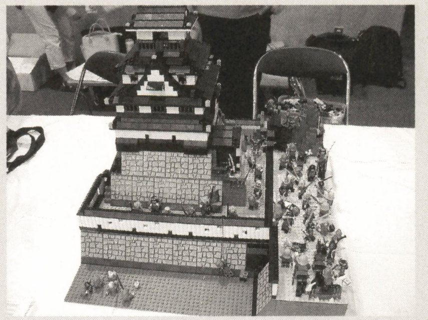
レゴ部門の入賞者		
1位	ピアノ 高橋 良一	写真8
2位	攻城戦! 相原 崇史	写真9
3位	スバル インプレッサWRC2001 けん・たっきい	写真10
4位	モーターショー二輪館 こみ	写真11
5位	MS-06R-1A　ザクⅡ ぐら	写真12

（敬称略）

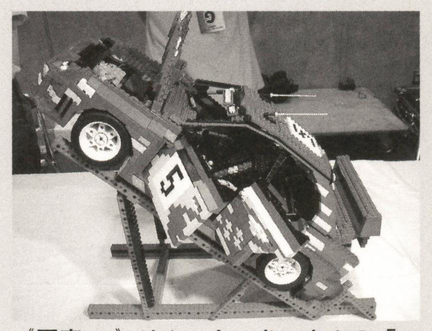
もっとも参加者が多かったレゴ部門です。テクニックシリーズのような作品や，お城シリーズのような作品，レゴトレインシリーズなどバラエティーに富んでいてとても見ごたえがありました。



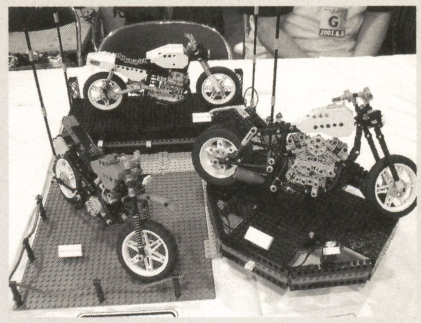
《写真8》高橋さんの「ピアノ」。鍵盤を押すと，中からいろいろなものが出てきます



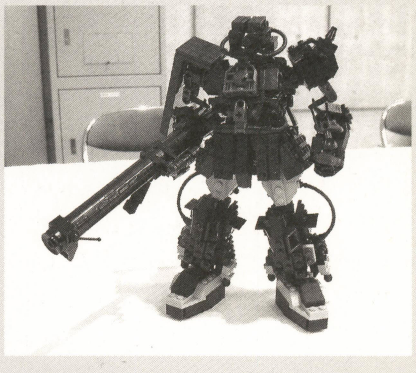
《写真9》相原さんの「攻城戦」。巨大なお城でみんなの目を引きました。フィグもたくさん



《写真10》けん・たっきいさんの「スバルインプレッサ」。実物と同じようにドアやボンネットが開きます。エンジンなども表現されています。これは，実車を8分の1スケールでレゴで作ったものです



《写真11》こみさんの「モーターショー二輪館」。3台のバイクが飾られていて，1台は普通に。1台はターンテーブルで回転しながら。もう1台はタイヤを回転させて展示されていました



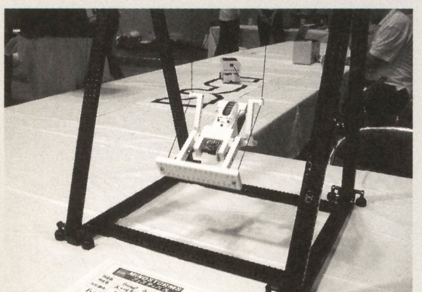
《写真12》ぐらさんの「MS-06R-1A ザクⅡ」。あのザクをレゴでうまく表現しています

MINDSTORMS部門

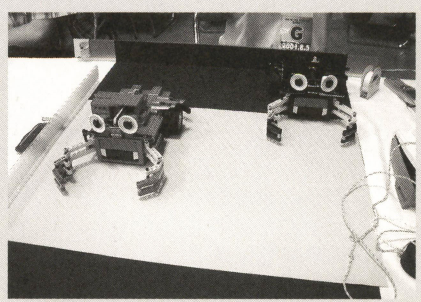
MINDSTORMS部門入賞者		
1位	Swing! Swing! Swing! 五十川 芳人	写真13
2位	ざりがにつりゲーム 東谷 賢一	写真14
3位	ライノ ピコ	写真15
4位	4×4ピックアップ・トラック おでん	写真16

（敬称略）

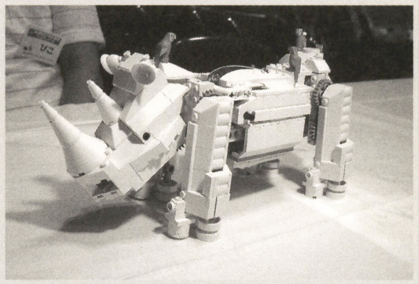
MINDSTORMS部門はRIS，RDS，DDK，DSDKの，コンピュータ部品を使っている作品のための部門です。今回の人気投票コンテストではとてもいろいろな使いかたをしていて，個人個人の表現のしかたが生きていたと思います。



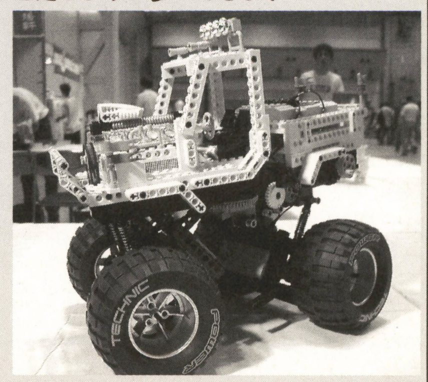
《写真13》五十川さんの「Swing! Swing!Swing!」。静止状態からこぎ始めて，かなりのところまでスピードに乗ってこぎます。とにかく，すごいとしか言いようがありません



《写真14》東谷さんの「ざりがにつりゲーム」。メッセージボールという赤外線を出す物をえさにして，ざりがにに見立てたロボットをつります



《写真15》ぴこさんの「ライノ」。白サイで4足速歩行ロボットです。どしどしと走るように歩いてきます



《写真16》おでんさんの「4×4ピックアップ・トラック」。現在販売されているレゴのタイヤの中でもっとも大きなタイヤを使って巨大な車を作っています。エンジンもピストンが動き，とてもかっこいいです

大型モデル部門

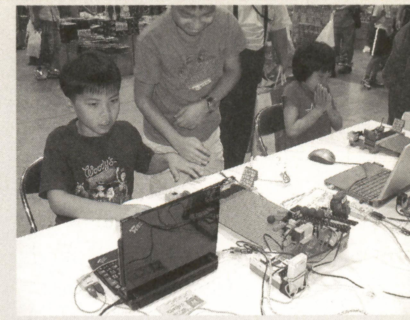
大型モデル部門入賞者		
1位	SILVER FALLS 伊藤 剛	写真17
2位	ネット対戦型 daisuke	写真18

（敬称略）

大型モデル部門は，あまりエントリーがなかったものの，作品ひとつひとつが大きいので，インパクトが強かったです。



《写真17》伊藤さんの「SILVER FALLS」。とても大きな岩山です。写真の人間と比べるとその大きさがわかるでしょう



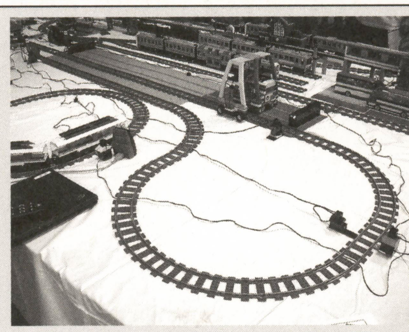
《写真18》daisukeさんの「ネット対戦型」。2台のパソコンでネットワークをつなぎ，LEGO CAM（レゴのカメラ）で画像を写しながら2人で遊びます。子供に大人気でした

審査員特別賞

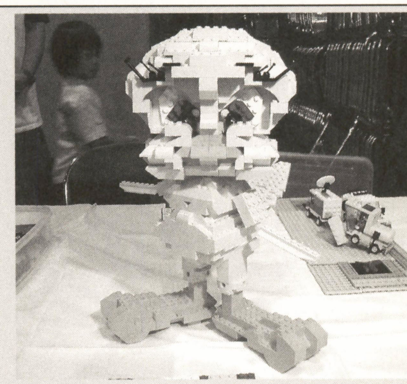
今回の人気投票コンテストには審査員特別賞がありました。これは，審査員の独断で「すごいな」と思ったものを選ぶものです。

審査員特別賞		
技術賞	自動車両連結器外し機と列車編成環状線 ザン ズンロン	写真19
芸術賞	カナリア いのすけ	写真20

（敬称略）



《写真19》ザンさんの「自動車両連結器外し機」と「列車編成環状線」の一部です。この作品はレゴトレインの車両の連結器を自動で見つけ，自動で切り離し，自動で走行させ，自動で編成の順番を変えるという優れものです



《写真20》いのすけさんの「カナリア」。モデルはアメリカのアニメのトゥイーティーです。レゴのパーツをものすごくうまく活用しています

その他気になった作品

表彰された作品を紹介してみましたが，ここからはボクが気になった作品を紹介しましょう。

Joeさんの「ロボットハンド」はしっかりと関節が動き，じゃんけんができるという，すばらしい“手”です。うまくすれば，物も握れそうな感じがします（**写真21**）。

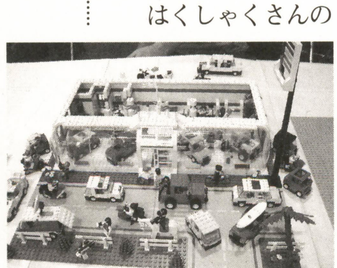
はくしゃくさんの「カーディーラー」はタイトルのとおり，車を売っているお店です。中の車はもちろん精巧にできていて驚きますが，外側の建物もガラス張りで雰囲気が出ていました（**写真22**）。

おでんさんの「消防車」は各部が動くようにできていて，遊んで楽しい作品になっています。全体的なバランスもよく，遊びがいもあるので，とてもよくできた作品だと思いました（**写真23**）。

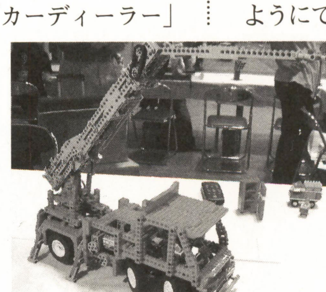
今回のフェスティバルは，競技会がなかったものの，展示だけでかなり盛り上がり，誰もが楽しめたと思います。やはり，これからもこういうイベントは続けてもらいたいと思います。



《写真21》Joeさんの「ロボットハンド」。手の関節がとてもよくできていて，物が握れそうです



《写真22》はくしゃくさんの「カーディーラー」。たくさんのフィグとたくさんの車が雰囲気を出しています



《写真23》おでんさんの「消防車」です。はしご車ではないのが特徴で可動部分が多く，とても遊びがいがありそうです

さて次回は !?

次回は，センサのちょっと進んだ使いかたを紹介します。

MINDSTORMSのセンサは組み合わせしだいでいろいろなものに使えます。それでは!!



最終回 Turbolinux 7 Workstationをインストール

ターボリナックス ジャパン

ターボリナックスのデスクトップ用Linuxの新製品「Turbolinux 7 Workstation」は，発売開始の9月7日(金)～9日(日)の3日間，全パソコンソフトの中でWindowsやMac用ソフトも抜いて売上げ第1位!*に輝き，全国主要パソコン・ショップでも現在，軒並み売れ筋上位にランクインされています。

ますます高まるLinuxへの注目もさることながら，日本語環境の充実，対応機器の豊富さ，使いやすいインストーラなどなど…幅広いユーザー層に支持される理由があります。

今回は，このTurbolinux 7 Workstationのイントロダクションとして，その魅力を紹介しましょう。

※東京秋葉原「T・ZONE.本店」，「LAOX ザ・コンピュータ館」，「ぷらっとホーム」調べによる



▲Turbolinux 7 Workstationのパッケージ
本体価格：15,800円(税別)
全国の販売店ならびにTurbolinuxオンラインショップで発売中。
http://www.turbolinux.co.jp/online_shop/
※大量導入される方には，お得なライセンスパックがあります。また，教育機関を対象とするアカデミックライセンスパックもあります。詳細はWebサイトで！

Turbolinux 7 Workstation魅力の全容
日本語デスクトップ用途に求められる機能・アプリケーションを充実

Windowsユーザーにも親しみやすく・使いやすい環境

Turbolinux 7 Workstationは，日本語デスクトップ環境の品質を最優先し，幅広いユーザー層を想定して，見た目や操作感などがWindowsユーザーにもなじみやすいものになっています。

Eメールをはじめとして，オフィスソフトやWebブラウザなどのデスクトップアプリケーションの充実はもちろん，USB機器対応やデジタルメディア対応の強化，ブロードバンド対応など，一般的なデスクトップ環境として十分な機能を提供します。

また，GUIインストーラによりLinux初心者でも導入しやすく，Linuxではじめて，一つのパスワードだけ入力すればインストールが自動的に完了する機能もあります。

マニュアルは全部で7種類。初級から上級までレベルに応じて段階的に利用できます。サポートは，期間中，件数無制限で受けることができます。

製品詳細は以下のアドレスで見ることができます。

http://www.turbolinux.co.jp/products/

【主要構成コンポーネント】
kernel 2.4.5，glibc 2.2.3，XFree86 4.1.0，rpm 4.0.2，gcc 2.95.3

【推奨環境】
PC/AT互換機
CPU：Intelx86互換 Pentium相当以上
メモリ：64MB以上(128MB以上を推奨)
ハードディスク：IDE / SCSI HDD 2GB以上を推奨
ビデオカード：VGA以上
(GUIインストール利用の場合にはXFree86 4.1.0対応のビデオカードが必要です)
その他：1.44MBフロッピードライブCD-ROMドライブ(ATAPI/SCSI) Ethernetカード

●対応ハードウエア情報は以下アドレスで
http://www.turbolinux.co.jp/support/tech/compatibility/compatible/

●ターボリナックスによる動作確認済みのPC一覧は以下アドレスで
http://www.turbolinux.co.jp/tech/compatibility/compatibility_pc.html

◆確認範囲
・インストール時のX Window Systemの自動認識と設定
・インストール時のサウンドカードの自動認識と設定(サウンド機能搭載機種のみ)
・インストール時のネットワークの自動認識と設定(ネットワーク機能搭載機種のみ)
※特記のないもの，その他のハードウエア(モデムなど)については動作確認を行っていません。

◆注意
ターボリナックスが行った検証結果に基づくもので，各ベンダーが動作保証するものではありません。各PCメーカーへの問い合わせはご遠慮ください。

その1　Windowsライクなデスクトップ環境

Turbolinux 7 Workstationでは，デスクトップのデフォルト環境に「KDE 2.1」を採用しました。これは，見た目やキーボードショートカットなどの操作感がWindowsユーザーにもなじみやすく，総合的に完成度が高いという観点からです。「GNOME 1.4」も搭載していますので，どちらでも選択可能です。



▲標準デスクトップKDE

その2　完成された日本語環境 WindowsでおなじみのリコーフォントとATOK X

Turbolinux 7 Workstationでは，Linuxとしてはじめて，リコーのTrueTypeフォント5書体をシステムフォントとして標準搭載しています。

リコーのTrueTypeフォントは，Windowsをはじめ，商用UNIXやPDAなどにも搭載され，実績，品質ともにNO.1です。アウトラインフォントに加えて，小さいフォントサイズでは視認性の高いビットマップフォントを採用することで，フォントサイズに依存しない美しい日本語表示が可能です。

※FTP版にもリコーTrueTypeフォントのTLゴシック（常用漢字版:2,719文字）が含まれていますのでその品質を確認することができます。

日本語デスクトップ環境のかなめである，かな漢字変換システムは，Windows，Macintosh環境で広く普及している，ジャストシステムの「ATOK X for Linux」を搭載。使いやすいインターフェースと高い変換効率で定評があり，Windows版ATOK辞書も使用可能です。

※書体サンプル：

TLゴシック

Tlgothic

TL明朝

Tlmincho

TL丸ゴシックM

TLMARUGOTHIC

TL教科書体

TLKYOKASYOTAI

TL行書体

TLGYOSYOTAI

その3　簡単インストーラ Mongoose

「Linuxのインストールは難しい」という従来の認識は，GUIインストーラの登場により払拭されはじめていますが，Turbolinux 7 Workstationでは，驚くほどインストールが簡単になっています。

たとえば…パスワード一つだけ入力すれば，あとは自動的にインストールが進行する「Turboインストール」。また，「ひとまずLinuxをためしてみたい」という方には，WindowsのFAT領域にインストールできる機能（ループバックインストール）がお薦めです。

さらに，パーティショニングによりインストールする場合には，インストールするハードディスクを設定するグラフィカルなツールもあります(TFDisk)。

Windowsなど他OSとデュアルブート環境で使用するためのプログラムとしては，おなじみのLILOブートローダーを使用しますが，ターボリナックスのLILOは，グラフィカルで直感的に改良しています。もちろん，CUIインストーラでも，GUIと同等の機能を提供します。

詳細は，本記事後半で，インストール設定画面を見せながら紹介します。

その4　USB機器を強力にサポート

Turbolinux 7 Workstationは，豊富なUSB機器をサポートします。これにより，キーボード，マウス，FD／CD-ROM／MOドライブ，スピーカ，Ethernetなど，周辺デバイスの選択肢が非常に広がっています。使用可能な各種デバイスを，インストール時点でインストーラが自動検出してくれることも大きな特徴で，各デバイスは電源を入れたまま取り外したり接続したりすることも可能です（ホットプラグ対応）。

さらに，ノート・パソコンでもLinuxを導入しやすいように配慮し，CD-ROMドライブが対応していない場合でもネットワークまたはローカルディスク経由でインストールできます。

また，一般的なモニタサイズはもちろん，携帯ノート・パソコン(1024×480，1280×600モニタ)でも，インストーラがビデオチップを自動検出して表示解像度を指定。ジョグダイヤルやモーションアイなどのノート・パソコン用ユーティリティもインストーラで選択して，スムーズに使用できます。

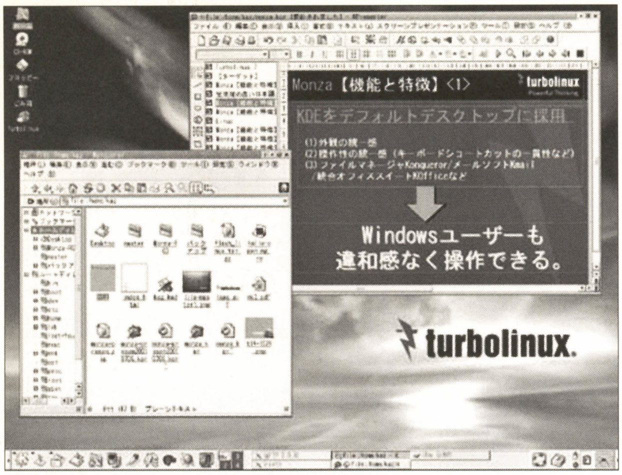
その5　充実したデスクトップ&機能

Turbolinux 7 Workstationでは，Webブラウザはもちろん，KDEのアプリケーションとして，メールソフト「KMail」，Webブラウザ・高機能ファイルマネージャ「Konqueror」，ワープロ/表計算/プレゼンソフトなどから構成される「KOffice」といった，十分実用に堪える日本語オフィススイートが充実しています。

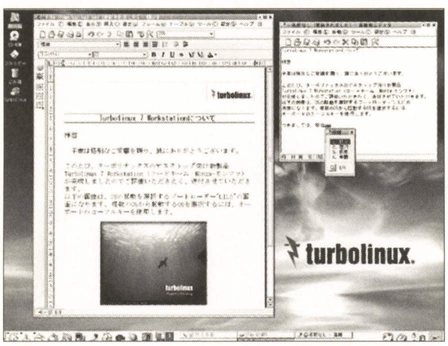
ブロードバンド時代の通信環境，最新デジタルメディア機能への対応も万全です。

◇充実したデスクトップツール・機能

・Webブラウザ…Mozilla(標準)，Netscape Communicator4.77，Konqueror，Galeon



▲kpresenter（プレゼンソフト）& konqueror（ファイルマネージャ）



▲kword（ワープロ）& Kwrite（高機能エディタ）

・オフィス&Eメール…KOffice/KMail, Evolution, Sylpheed
・ADSL接続…設定ツールで簡単
・PDAと同期接続…スケジュール管理の効率アップやデータのバックアップなどが可能
・画像/映像取り込み…デジタルカメラ，ビデオデータ
・サウンド再生・編集…MP3，Ogg Vorbis，MIDI，RealAudio

その6　充実したマニュアル 件数無制限サポート

製品に付属するマニュアルは，全部で7種類。

・Linuxベーシックガイド
・インストールガイド
・ユーザーガイド
・アプリケーションガイド
・コマンドガイド
・サポートガイド
・FAQ

インストールガイドやユーザーガイドはもちろんのこと，Linuxベーシックガイドやコマンドガイドなどを完備していますので，初級者にはお薦めです。

Linuxベーシックガイドで知識をつけて，ユーザーガイドでデスクトップ操作，GUIアプリの使用方法をマスター。さらに，より幅広くLinuxを活用していただくためにコマンドガイドで，Linux上級者へとステップアップしてください。

製品購入後のサポートサービスは，従来のインストールサポートに加えて，サウンド設定やUSB機器接続まで対象範囲を拡大し，期間中は，件数無制限で問い合わせに対応しています。

【バンドルする主なソフトウエア】

●商用ソフトウエア

・日本語変換システムATOK X for Linux
・リコーTrueTypeフォント5書体

●Java開発環境

・Sun JDK1.3.0
・Sun Forte for Java, Community Edition 日本語版

●その他開発環境

・Borland Kylix 1.0 Open Edition
・各種コンパイラ/デバッガ（gcc，g++，g77，gdb，ddd）
・各種スクリプト言語（python，perl，ruby，php）
・各種統合開発環境IDE（Qt Designer，Anjuta，Kdevelop，Guikachu）

●Netscape・Mozillaプラグイン

・RealPlayer 8 Basic
・Adobe Acrobat Reader 4.05

●プリンタフィルタ

・Canon Bubble Jet Print Filter for Linux
EPSON Photo Image Print System / GS Laser Printer Driver

簡単インストーラMongoose
Windows環境はそのままに，ループバックインストールで，ひとまずTurbolinuxにチャレンジ！

新インストーラがいかに簡単かを紹介する前に，「Linuxにはじめて挑戦してみよう！」という人に最適な方法として，ループバックインストールを紹介しましょう。

この連載の第1回で紹介したように，

(1)Linux用に新しいPCを用意する
(2)Linux用に別のハードディスクを用意する
(3)一つのハードディスクにLinuxとWindowsの両方をインストールする

など，Linuxのインストール方法には，いくつかの方法があります(2001年4月号参照)。

ループバックインストールは，Windowsで使用している既存のPCに必要な空き容量さえあれば，Windows環境を消去せずに，Linuxをインストールでき，完全にWindowsとLinuxとのデュアルブート環境を構築できるのです。

なにはともあれ，今はLinux専用のPCまたはディスクを確保できないけれど，ひとまずTurbolinuxを試してみたいという方は，ぜひこの方法でトライしてみてください。

ただし，(1)～(3)の方法でインストールする場合と異なり，Linuxを起動する際は，必ずフロッピーブートすること，Linuxシステム内でのパーティションの分割ができないこと，2GB以上のLinuxパーティションを作成できないことなどの制約があります。

簡単インストーラMongoose
GUIでの最短コース

それでは，Mongooseインストーラのいくつかのステップを追いながら，Turbolinuxのインストールがいかに簡単かを紹介しましょう。

MongooseのGUIインストーラでは，左側に，インストールの項目と進捗状況が表示されますので，とっても便利です。さらに「F1」キーを押すと，それぞれのヘルプが参照できます。

また下のほうに「中止」ボタンがありますので，「やっぱり明日にしよ…」ということになったら，いつでもインストールを止めることができます。

インストーラの詳細は以下でも見ることができます。

http://www.turbolinux.co.jp/press_room/material/mongoose/

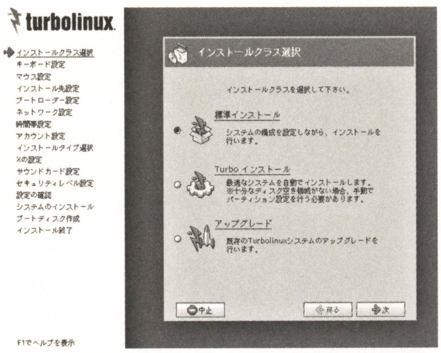
●インストールCDから起動した場合の最短コース

インストーラの起動
↓
言語の選択
↓
GUI/CUIインストーラの起動
↓
インストールクラスの選択
↓
キーボード設定
↓
マウス設定
↓
インストール先設定
↓
ネットワーク設定
↓
時間帯設定
↓
アカウント設定
↓
インストールタイプ選択
↓
Xの設定
↓
サウンドカード設定
↓
セキュリティレベル設定
↓
設定の確認
↓
システムのインストール
↓
ブートディスク作成
↓
インストール終了

インストールクラスの選択

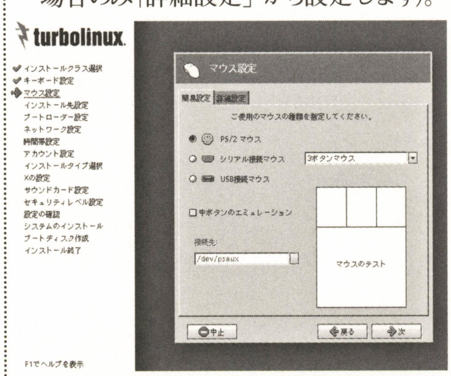
「標準インストール」「Turboインストール」「アップグレード」の三つの選択肢がありますが，なんといっても初心者にお薦めなのが「Turboインストール」。

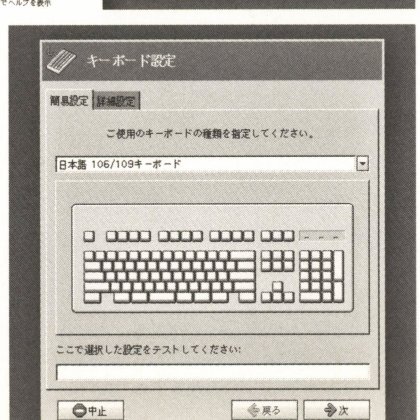
これは，rootのパスワードを設定するだけで，ほとんどの設定をインストーラが自動で設定してくれます。



キーボードやマウスの設定

使用するキーボードや，マウスの種類を選択しますが，多くの機種が，プルダウンメニューから選択できるようになっています。マウスの種類は，インストーラが自動検出したマウスが選択されています（リストに含まれていないものを使用する場合，自動検出されなかった場合のみ「詳細設定」から設定します）。

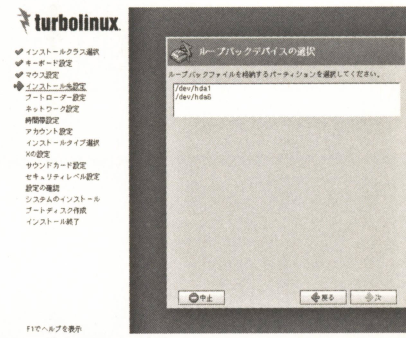


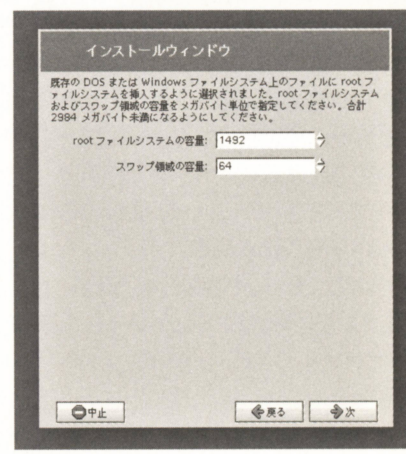


インストール先設定

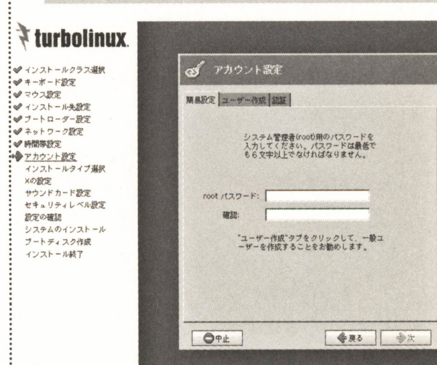
「自動パーティション設定」「TFDisk」「ループバックインストール」の三つの方法を選択できます。

ループバックインストールを選択すると，既存の空き領域が一覧表示されます。その中からインストール先を選択して,容量を指定します。スワップ領域は，128MB以上は確保してください。





アカウント設定

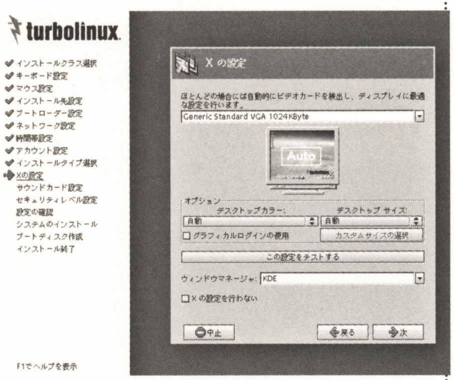


インストールタイプ選択

インストールするパッケージを選択します。「標準選択」では，「標準デスクトップ」「ラップトップPCデスクトップ」「開発ワークステーション」などの，用途に応じて最適な内容を選択できます。

Turbolinux 7 Workstationには充実した開発言語や，ライブラリなどが収録されていますが，これらを必要とする場合は，「開発ワークステーション」を選択してください。

デスクトップPCに最適な「標準デスクトップ」や，ノートPCに最適な「ラップトップPCデスクトップ」では，開発ツールや，サーバー用パッケージはインストールされません。



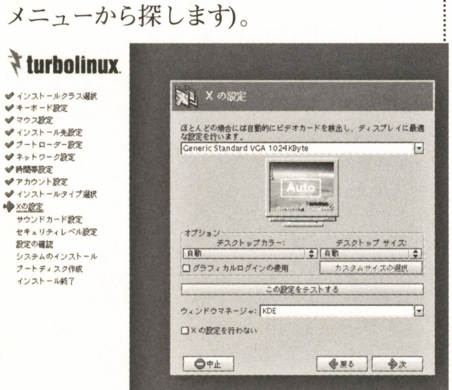
Xの設定

X Window Systemを利用するために，モニタとビデオカードの設定をします。

従来「これが難しい」と言われていた難所のひとつですが，Turbolinux 7 Workstationでは，「画期的に簡単になった!!」というユーザーの声が多くあります。

モニタの種類は，インストーラが自動検出して，「製造元」と「モデル名」をあらかじめ選択しています。

ビデオカードも同様に，インストーラが自動検出して選択されています(自動検出されなかった場合のみ，プルダウンメニューから探します)。

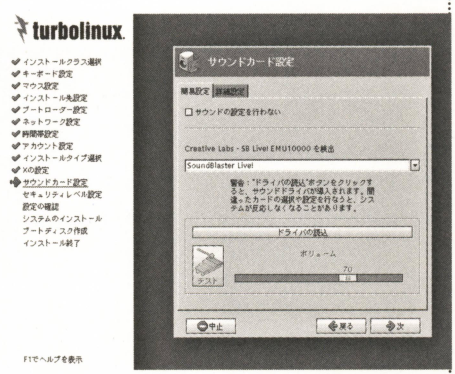


これでもう完了です。設定が終了したら，「この設定をテストする」を押して，X Window Systemの動作を確認してください。



サウンドカードの設定

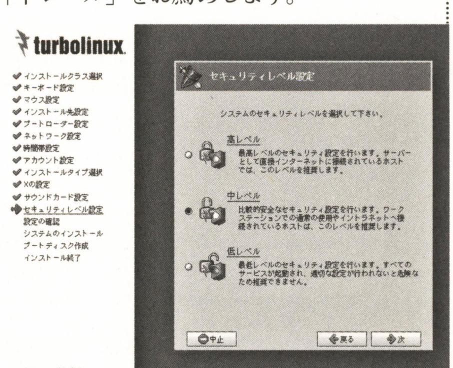
サウンドカードもインストーラが自動検出し表示しています。



セキュリティレベルの設定

セキュリティレベルの設定は，主にネットワーククライアントとして使用する場合に関係してきますが，Linuxが提供するさまざまなサービスの中で，実際に使用しないものを起動させておくと，システムへの不正侵入などの原因となる場合があるので，用途にあったレベルを選択します。

「高」「中」「低」の三つのレベルから選択できますが，クライアント用途には「中レベル」をお薦めします。

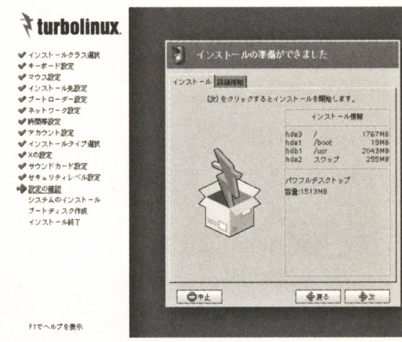


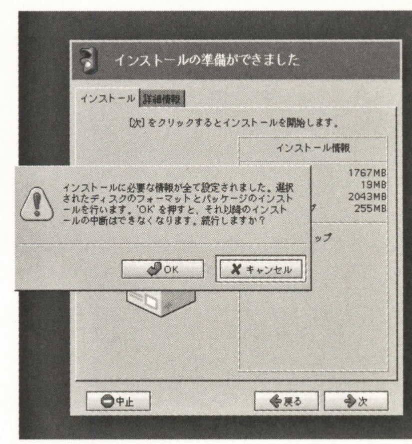
設定の確認

インストール作業で設定した内容を確認できます。

再設定したい場合は，「戻る」を押すと，それぞれの設定画面まで戻ることができます。

システムのインストールを確認するメッセージが出て「OK」を押すと，あとは，インストールが完了するのを待つばかりです。





最後に

さあ，今すぐTurbolinux 7 Workstationをインストールして，これまで「よいこのLinux修行」で紹介したことを，自分のパソコンで実行してみてください。

…ちょっと，製品PRになってしまいましたが，今回の製品は，開発を担当したターボリナックスのエンジニア陣も「Turbolinux有史以来の最高の出来!!」と自負しており，これまでβ版から厳しい目で検証していただいた，多くの既存ユーザーさんからも「今度のTurbolinuxはなかなかいいじゃないか!!」と，絶賛されている自信作なのです。

さて…最後になってしまいましたが，今回をもって「よいこのLinux修行」は，一旦お休みさせていただきます。またお目にかかれるのを楽しみにしています。長い間ありがとうございました。

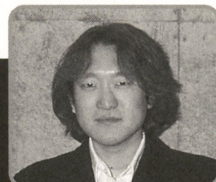
DTM Art Studio
パソコンで音楽をたのしもう!

協 力
ヤマハ（http://www.yamaha.co.jp）

Text&SoundData
仲野方敏／上村学

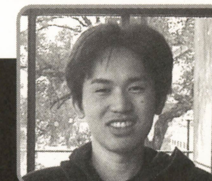
このコーナーは，芸大生が体験したDTMの基本，応用，実践を紹介するコーナーです

筆者紹介



仲野方敏：
大阪芸大
芸術学部　音楽学科
音楽工学コース（音楽工学専攻）3回生

みなさんこんにちは，仲野です。早いもので，この連載もすでに7回目。みなさんの音楽に関する知識は増したでしょうか？まあ，どちらにしても，何事も続けることが大事です。作曲も同じことで，いい曲ができないからといってすぐに諦めてしまってはいけません。繰り返し作曲していくことで，良い曲を作ることが可能になるのです。というわけで今回もがんばっていきましょう。



上村学：
大阪芸大
芸術学部　音楽学科
音楽制作コース（作曲専攻）3回生

こんにちはみなさん！「らくらく作曲名人2」の使いかたはだいぶ慣れてきたでしょうか？音楽を自分で作ることのおもしろさがたくさんの人に知ってもらえることができたらいいなぁと思いながら原稿を書いている今日この頃です。音楽を作りだすことは大変なことです。しかし苦労して作った曲をほめられると本当に気持ちいいものです。さあ今回もそのための勉強を始めましょう。

YAMAHA「らくらく作曲名人2」
～アレンジ編　その2～

先月号のおさらい

前回はモチーフ（メロディ）を音符入力し，そのメロディに伴奏をつけるアレンジをしましたね。音符入力がまだむずかしい人やわからない人は，ホームページにあるメロディのテンプレートのデータを，らくらく作曲名人2に読み込ませることで音符入力をはぶきました。

アレンジは，ステージ画面左側のカーテンにあるスタイルパレットからスタイルを選んで，オートアレンジボタンで伴奏をつけました。そして，その伴奏のコードとセクションを編集しました。コードの編集は知識のある人以外は少しむずかしかったかもしれませんが，おすすめのコードを選べる機能を使えば，多少は編集できたんじゃないでしょうか？セクションの編集のほうはコードより簡単にできたと思います。

「らくらく作曲名人2」

「らくらく作曲名人2」は，音楽の専門知識がなくても，作曲・編曲等が簡単にできるるソフトです。「めちゃらく作曲名人」とちがい，シーケンサとして使うこともできるので，DTM初心者以外にもお勧めです。それに，楽譜を制作することも可能なので，多くの人が楽しめるソフトです。

製品仕様
- CD-ROM（「らくらく作曲名人2」，ソフトシンセサイザ「S-YXG50」，ヤマハCBXドライバ，MidRadioプレーヤ，用語集）
- 取扱説明書

動作環境
- 対応パソコン：NEC PC-98シリーズ，IBM-PC／AT並びにその完全な互換機
- 対応OS：WindowsMe/98/95/NT4.0/2000
- CPU：Pentium-166MHz以上/Celeron以上（ソフトシンセサイザ同時使用の場合MMX Pentium-233MHz／Celeron以上推奨，Windows2000の場合Pentium II 300MHz以上推奨）
- 必要メモリ：32MB以上（64MB以上推奨）
- ハードディスク必要容量：40MB以上
- ディスプレイ：800×600ドット，24ビットTrue Color以上
- 音源：ソフトシンセサイザまたはXG対応MIDI音源機器
- 必要機器：マウス，ハードディスク，2倍速以上のCD-ROMドライブ



14,800円

まあ，むずかしくなってきたぶん，自分で変更できる部分が増え，個性的な作品が作れるようになってきたと言えます。さぁ今回も，がんばってアレンジに挑戦しましょう！

さらに進んだアレンジに挑戦！！

前回までは，アレンジに必要ならくらく作曲名人2（写真1）の使いかたを説



《写真1》らくらく作曲名人2の画面

明してきたんですが，今回はより音楽的なアレンジを説明したいと思います。また，今回も筆者が作った曲の中から「BLUE」という曲をモチーフ（メロディ）に使いたいと思います。

さらに進んだアレンジに挑戦ということで，みなさんがアレンジをするときに役立つと思われることを中心に，いくつかのジャンルを取り上げ，それぞれのジャンルについての音色，リズム，コード進行などについて説明していきたいと思います。
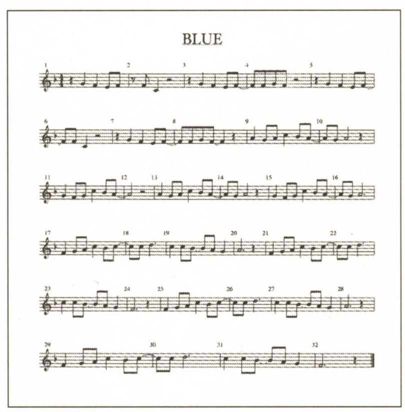
また，らくらく作曲名人2を卒業して他の音楽用ソフトを使いはじめるときにも役立ててもらえるよう，より広い視野でそれぞれのジャンルについて説明しています。もちろん，らくらく作曲名人2を使ってのジャンル別アレンジ方法のコツやヒントも解説します。

さらに，筆者ならこんなやりかたでアレンジをするという，らくらく作曲名人2を使ったアレンジ方法の一例も紹介します。それではじめましょう。

音符入力

今回も前回と同じく，**写真2**の楽譜を見ながら音符を五線譜画面に入力していきましょう。今回はタイがたくさんでてきます。入力のコツとしては，同じ高さの音符はボンドツールで両方をクリックすることで，二つの音符をつなげることができます。

まだ音符入力ができない人や，むずかしいという人のために，ホームページにメロディのテンプレートがあるので，そちらを使ってもOKです。音符入力がすんだら，さっそくアレンジをはじめましょう。



《写真2》使用する楽譜はこれ！

～風～っぽく変化させるアレンジの極意

ロック風アレンジ

ロック風にアレンジする場合についてですが，まず音色について説明すると，やはり，ディストーションもしくはオーバードライブなどを使って歪ませたギターサウンドが特徴といえます。最近では，一口に「ロック」といっても，さまざまなジャンルがありますが，どのジャンルにしても，こういったギターサウンドが要であると思います。他に，ハモンド・オルガンの音色を入れることも，ロック風にアレンジする際の有効な手段だと思います。らくらく作曲名人2を使ってアレンジするときは，音色選択画面（ステージ上のプレーヤーを右クリックし，サブメニューから選択）を使って設定するとよいと思います（**写真3**）。

その他の特徴としては，バンドとしての形態をとったものがほとんどなので，音の数（パート数）はそれほど多くありません。ロックバンドのメンバー写真などを見てもらえればわかると思いますが，ほとんどの場合，3人～5人程度のメンバーでバンドを組んでいます。

構成としては，ボーカル，ギター，ベース，ドラムという編成が多いのですが，ギターを2本使っていたり，キーボードが入っていたりすることもありま

民族音楽のスタイル

南アジアの音楽

南アジアという地域は，インド，パキスタン，バングラデシュ，ネパール，ブータン，スリランカ，モルディブの7か国で構成されています。

インドの音楽といえばほとんどの人がシタールの音を想像すると思います。シタールの音はMIDI音源の中にも入っているので「らくらく作曲名人2」でももちろん使うことができます。

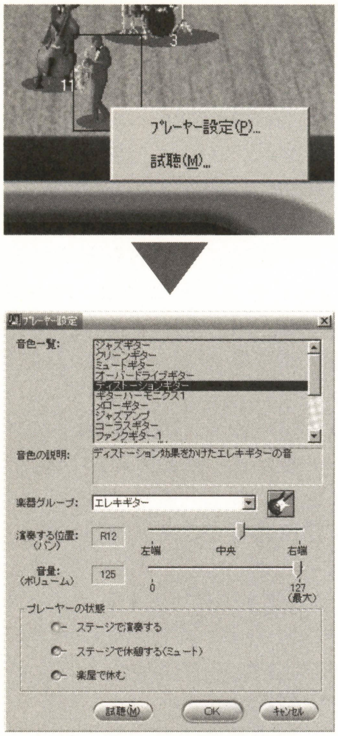
このシタールという弦楽器と打楽器のタブラーの伴奏をともなう音楽がインドの音楽だと思っている人がほとんどだと思います。しかし，シタールとタブラーの音楽は北インドの音楽であり，南インドではあまり人気がありません。また，南北ともに主な音楽は声楽であって，器楽ではありません。つまり，私たちが想像していたインドの音楽はインド音楽のほんの一部であり，主流となる音楽やその他にもたくさんの音楽が存在しています。

そこで，これらインドの古典音楽に使われる理論である「ラーガ」と「ターラ」について説明したいと思います。南インドと北インドの音楽では地域的，言語的，歴史的に異なるため，表面上は，音楽理論や楽器にちがいがあります。しかしこれら北と南のインドの音楽には，共通の側面もあります。それが「ラーガ」と呼ばれる旋律の規則と「ターラ」と呼ばれるリズム周期の理論です。

「ラーガ」という規則には，西洋音楽のような階段状の一定間隔の音階ではなく，音と音の間には無限の間隔の音高が存在し，ある特定のルールを守りながら駆使するものです。それぞれの音の性格や扱いかた，特徴的なフレーズによって，特定のラーガが性格づけられます。西洋音楽と同じ音高の7音音階を基本とするラーガやジグザグ進行するラーガなど数百種類のラーガが存在します。

また，「ターラ」は2拍，3拍，4拍，5拍など拍の固まりをいくつか組み合わせて一周期とするもので，概念的には西洋音楽の拍とはちがいます。音楽は必ずしもターラの1拍目から始まるとはかぎりません。しかし最後は，1拍目にもどって終わらなければならず，周期の途中で切れてしまってはいけないという決まりもあります。ターラは，しばしば非常に複雑な構造をしていて，一周期が20拍以上になることもあり,その複雑さと正確さの度合いが，演奏家の技量を示す指標になっています。

南アジアの音楽を代表してインドの音楽について説明してきましたが，インドの音楽も，理論がわかれば楽しみかたもちがってくるのではないでしょうか？



《写真3》音色選択画面

す。これは，全体のサウンドを，よりヘヴィーなものにするためや，ハーモニーの響きをより重厚かつ壮大なものにするためなど，さまざまなねらいがあります。年代的に見た場合には，現代的なものになるほど，バンドの構成人数が多くなる傾向が見られます。

つぎに，リズムとコード進行についてですが，ロックミュージックのリズムとコード進行は，比較的シンプルなものが多いようです。個人的な感想を述べさせてもらえば，複雑にすればするほど，軽い感じ（ポップスのように）になってしまうように思います。もちろん，すべてがそうなるわけではありませんが，他のジャンルの音楽に仕上げるときにも役立つと思うので，より簡単に重い感じを出すための手段として覚えておくといいでしょう。

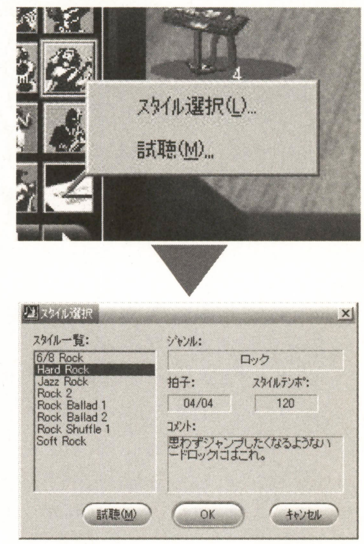
具体的に説明すると，ドラムやベースの激しい動きをなくす（同じリズムの形を刻み続けるなど）ことなどがその方法として挙げられます。

アレンジという話題から少しそれてしまいますが，各パートの音量のバランスについても触れておきましょう。ロック系の音楽を他のジャンルの音楽と聴き比べてみるとわかるのですが，ロック系の音楽では，それぞれに若干のちがいはあるものの，ドラム（とくにスネアドラムやキック）やベースの音を強めに出す傾向があります。こうすると，アレンジした曲をよりロック・ミュージックらしく聴かせることができます。

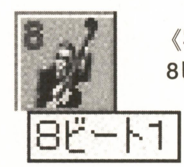
らくらく作曲名人2でロックっぽくアレンジするためのコツは，ステージ画面のステージ左側のカーテンにあるスタイルパレットからロックのアイコンを左クリックして（**写真4**），［スタイル選択］からスタイルを選択するというのが一般的ですが（**写真5**），そのほかにも8ビートパレット1（**写真6**）の［スタイル選択］から「Pop Rock1」や「Pop Rock2」を選んだり，8ビートパレット2（**写真7**）の［スタイル選択］より「8Beat Heat」を選んでもいいと思います。その他にも探せばロックっぽく作れるスタイルはあると思いますが，一見ロックっぽく聞こえないスタイルでも音色やテンポを変えることによってロックぽくなったりします。



《写真4》
ロック・アイコン



《写真5》スタイル選択画面



《写真6》
8ビートパレット1・アイコン



《写真7》
8ビートパレット2・アイコン

そして［オートアレンジ］ボタンをクリックしてオートアレンジするときは，曲調確認画面の［標準］か［シンプルに］を選ぶと，それらしくなります。もちろんそれだけではおもしろくないので，

シリーズ音響学

耳

今回のシリーズ音響学は，人間の耳について進めたいと思います。普段の生活の中で，みなさんが自分の耳について考えることはあまりないと思います。しかし，よくよく調べてみると，我々の耳には驚くべき特徴や機能が備わっています。

まずは，それらについて説明する前に耳の仕組みについて説明しておきましょう。耳は，大きく三つの部分に分かれています。耳の穴に近いほうから，「外耳」，「中耳」，「内耳」と呼ばれます（ちなみに「外耳炎」とか「中耳炎」などの「外耳」，「中耳」とはまさしくこのことを指しています）。

音波が耳に到達すると，外耳道の空気が振動し，鼓膜を振動させます。そしてその振動が，中耳にある聴小骨（つち骨，きぬた骨，あぶみ骨の三つ）によって，約50倍の圧力変化として内耳の蝸牛殻に伝わります。蝸牛殻はリンパ液で満たされていて，この中を伝わる振動が聴細胞を刺激し，その信号が大脳に伝わって音として知覚されます。

それでは，つぎに耳の物理的な特性についてです。人間の耳は，理論上では20Hz（ヘルツ）～20KHz（キロヘルツ）の音しか聴くことができないとされています（CDの規格はこのことに基づいて決められています）。この周波数間の音，つまり耳で聴くことのできる周波数を「可聴周波数」と呼びます。ただ，あるレベル以上の刺激量がないと，鼓膜が振動しないので，可聴周波数の音でも音量によっては必ずしも聞こえるわけではありません。また，外耳道の特性から，人間の耳には「2KHz～2.8KHz」が最もよく聞こえます。

つぎに，音の大きさについてですが，感覚量としての「大きさ」は物理量としての「強さ」だけでなく，周波数の影響も大きく受けます。つまり，物理的に同じ大きさの音を聴いても，周波数によって

●用語解説
ディストーション：エフェクタ（音を加工して何らかの効果を得るための機器）の一種で，音響機器に過大入力を与えたときのような音の歪み（割れ）を意図的に作り出す機器。オーバードライブもほぼ同じ効果を得るためのものだが，ディストーションによって得られる歪みが鋭角的かつ金属的なものであるのに対して，オーバードライブではよりマイルドで温かみのある歪みを得ることができる（歪みの度合いが低い）。ロックギターのサウンドメイクには必ずと言っていいほどこれらが使用されている。

コードの編集もしないといけないんですが，あまり複雑なコードに差し替えないほうがよいでしょう。

ユーロビート風アレンジ

ユーロビートは，なんといっても4ビートでのバスドラムの刻みが特徴です。ユーロビートの曲では，他のジャンルでは考えられないくらいバスドラムの音を前面に押し出します。このリズムの形は，ユーロビートではかなり多用されているものです。らくらく作曲名人2，もしくは他の音楽用ソフトを使う場合にも言えるのですが，バスドラム専用にトラック（パート）を作ってしまうと，あとでミックスするときにやりやすくなります。

その他の特徴としては，ほとんどの曲のテンポがアップテンポで，曲によって多少のちがいはあるものの，「4分音符＝100」以上のテンポで作られているようです。また，音色についてはシンセサイザで作られた機械的な音色以外はほとんど使われておらず，ギターの演奏等が入っていたりしても，エフェクトをかけて機械的なサウンドに仕上げられています。これは，サウンド面での全体的な調和を図るためだと思われます。ちなみに，ボーカルにもほとんどの曲でエフェクト処理が施されています。

つぎに，演奏についても触れておきましょう。こういったジャンルの作品を聴く限り，ほとんどのパートが打ち込みによって作られているように感じます。そのため，カラオケそのものは非常に機械的な印象を受けます。しかし，逆に言えばこの機械的な感じこそがユーロビートの特徴であるとも言えます。

また余談になりますが，ユーロビートという音楽を知らないという人は，ダンスの「パラパラ」のバック・ミュージックとして使われている音楽を思い出してみてください。テレビなどで一度ぐらいは聴いたことがあるでしょう。ユーロビートとはまさしくあの音楽のことです。

らくらく作曲名人2でユーロビートっぽくアレンジするためのコツとしては，スタイルパレットからダンスフロアアイコン（**写真8**）を右クリックして，スタイル選択画面からいずれかのスタイルを選択すると，それっぽいスタイルになります。オートアレンジのほうは曲調確認画面で［標準］［明るく］［シンプルに］［暗く］等を選べばよいでしょう。



《写真8》ダンスフロア・アイコン

しかし，このままではあまり変化がないので，コードの編集で，ところどころに変わったコードを入れるなどしたら，おもしろくなると思います。

ポップス風アレンジ

ポップスとは，ポップな音？なんでしょうか。辞書で調べるとポピュラーミュージックのこととか，軽妙で洒落ているさまとか，大衆的で商業性のあるさまと書かれています。つまり聴きやすくて，みんながわかる曲というジャンルなんでしょうか？最近の日本のポップスもいろいろ多様化しています。アイドルが歌っている曲やバンド形態で演奏したり，打ち込みの音楽で歌われたりしています。またロックとポップスの間のような音楽もあります。ポップスとは非常に幅の広い表現であいまいなものになっています。わかりやすいジャンルで，しかも一番身近にあるのにポップスってなに？と考えるとむずかしいものですね。

しかし，ポップスにも全体的な共通点はあります。キャッチーであったり，メロディアスであったり，サビの部分が強調されていたりします。ということはポップス風にアレンジをしたいのであればメロディを強調したり，Aメロ，Bメロ，サビ等をわかりやすくして聴きやすいアレンジにしたらいいでしょう。

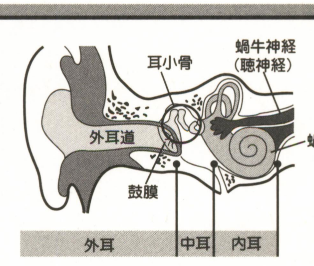
らくらく作曲名人2でポップス風にアレンジするには，スタイルパレットからJポップアイコン（**写真9**）を左クリックして，スタイル選択画面の中からいずれかを選択するのが一般的ですが，それ以外にも8ビート1，2アイコンの中

人間の耳にはちがった大きさに聞こえるということです。以下にそのことに関する法則を簡単に説明しておきます。

①同じ音圧レベルでは低音（低周波数）より高音（高周波数）のほうが聞き取りやすい
②音圧レベルが小さいときには，低音と高音は聞き取りにくい
③音圧レベルが大きくなると周波数による差は小さくなるが，音圧レベルが小さくなると差が大きくなる。

それでは，最後にマスキング効果というものについてです。マスキング効果とは，聴こうと思う音がそれ以外の音（妨害音）によって聴きにくくなる現象です。この現象は，視覚や味覚にもみられますが，音に関してはつぎのことが言えます。

①音量の差が大きくなるとマスキングの効果も大きくなる
②二つの音の周波数が近いほどマスキングされやすい
③周波数が離れている場合は，低音は高音をマスキングしやすいが，高音は低音をマスキングしない
④時間的に先になった音が，後の音をマスキングすることもある。



最後にこの現象を説明したのは，ミックスをする際に役立つと思ったからです。ミックスをしていて「どうしてもボーカルが他のパートに負けて埋もれてしまう」というような問題が発生することがあります。そういったときに，この現象のことを知っていると，問題を解決する糸口になると思います。ぜひ覚えておいてください。



《写真9》
Jポップ・アイコン

から選択してもいいでしょう。また，オートアレンジするときの曲調確認画面では，上の段の［標準］［明るく］［シンプルに］［暗く］などを選んだほうがいいと思います。あまり複雑なコードを使うと聴きにくくなりますから。

とは言ってもこのままではおもしろみがないので，あとでコードを部分的に複雑なコードと差し替えてみるのもいいかもしれません。

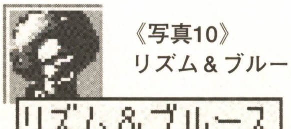
そしてセクションの編集ですが，前にも書いたようにAメロ，Bメロ，サビ等をわかりやすくするために，それぞれの間にFillを入れてわかりやすくして，とくにサビの前では少し盛り上げ，サビでさらに盛り上がるようにするとポップスっぽくなると思います。

R&B風アレンジ

R&B（リズム＆ブルース）は最近の日本のポピュラー音楽業界でも人気が出てきていますよね。では，R&Bとはどんな音楽なんでしょうか？ブルースはもともと弾き語りスタイルが基本です。そこにリズムを新しく加えた黒人音楽全般をR&Bといいます。

アフリカ系アメリカ人のテイストを持った楽曲のほとんどが，R&Bスタイルと言っても過言ではないかもしれません。私が思うには，アフリカ系アメリカ人はリズム感がよくてリズム中心の音楽だと思います。一般的に弱拍とか裏拍とか言われている4／4拍子でいう2拍目4拍目を揺らして少し遅らせたり，早めたりしてリズムをはねた感じにしたり，ためた感じにしたりすることで，音楽そのものをゆらして独特のグルーブを作っています。

楽器の特徴としてはやはりドラムが個性的で目立って聞こえます。また，ベースの音がメロディックに動いたりすることもあります。らくらく作曲名人2でR&Bのようにアレンジするときは，スタイルパレットからリズム＆ブルースアイコン（写真10）を左クリックして，スタイル選択画面の中から選ぶのもいいんですが，今日本で流行っているR&Bよりブルースっぽいスタイルが多いので，ゆっくりめの曲にしたいのだったらバラードアイコン（写真11）からModernを選択してもいいかもしれません。



《写真10》
リズム＆ブルース・アイコン



《写真11》
バラード・アイコン

オートアレンジをするときの曲調確認画面では，R&Bは複雑なコード進行のほうがそれらしくなるので，下の段の［スリリングに］［ジャズっぽく］［斬新に］［渋く］の中から選んでもいいですが，上の段の基本的なコード進行を選んでおいて，あとで全体的にコード編集で複雑なコード進行に変えていってもいいと思います。

芸大生ならこうやる！

●仲野

今回のテーマがアレンジということなので，僕達もみなさんと同じモチーフを使ってアレンジしてみました。

僕のほうは，今回のモチーフをロック風の曲に仕上げたつもりです。まず，メロディ（モチーフ）を打ち込んで，アレンジの作業に移りました。アレンジにおいては，曲そのものの感じを重くするために，できるだけシンプルにしました。しかし，メロディから受ける印象では，極端に重くすると変な曲になってしまいそうだったので，少しだけはねた感じが出るようにリズムの形を考えました。

ちなみに，この曲ではオートアレンジ機能も使っています（写真12）。



《写真12》オートアレンジ機能

音色とパートに関しては，できあがったときの音の細さを補うために，ギターパートを三つにしています。それぞれのギターの音色はすべて同じ音色（ディストーションギター）にしてあります（音色選択画面を使用）。それに加えて，全体の明瞭度を増すためにストリングスの音を少しだけ聴こえるようにしています（写真3参照）。

ミックス（各パートの音量バランス）については，とにかく，重い感じの曲にすることを第一に考えて作ったので，ギターとベースを強めに出すようにしました。しかし，ドラムが弱くなると全体のノリが悪くなるので，ギターとベースがドラムに勝ってしまわないように気をつけました。ストリングスは，前述のように明瞭度を増すためだけのものなので，聴こえるか聴こえないかという程度しか出していません。

その他には，スタイルを「ハードロック」に，ホールタイプを「ステージ1」にしました（写真13）。

みなさんも，僕達の作品以上のものをがんばって作ってみてください。



《写真13》完成画面（仲野）。「ハードロック」スタイルで，ホールは「ステージ1」

●上村

僕もアレンジしてみました。とは言ってもこの曲は僕が作ったものなので，一度はアレンジしているんです。あらためてらくらく作曲名人2でアレンジしたらどんなアレンジができるのか期待に胸をふくらませながらやってみました。今回，僕自身も個人的に好きな打ち込み系ポップス風にアレンジしようというヴィジョンを持って取り組みました。

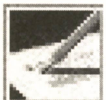
まず，メロディを入力して，スタイルパレットの中からディスコアイコンを左クリックしてDisco Popを選択しました。そしてアレンジボタンを押して曲調確認画面で［明るく］を選択しました。つぎにコードの編集です。コードは僕が作曲したときに付けたコードが気に入っているので，そのコードにすべて差し替えました（写真14）。とは言っても僕が付けたコードと同じところも結構あって，僕とらくらく作曲名人2は似



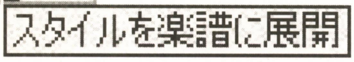
《写真14》コード差し換え

てるのかなぁ？と思ってしまいました。

その後はセクションの編集です。Fillを少し付け加えたりしました。その後は一度このアレンジを［楽譜に展開］ボタンで楽譜に展開しました（**写真15**）。ここで，このデータを保存しました。

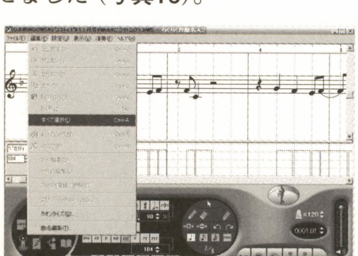


《写真15》楽譜展開ボタンは，スタイルパレットの一番下の一番右にある



なぜここで楽譜に展開して保存したかというと，ここでアレンジしたデータのTr.12（プレーヤーの足下番号12），Tr.14（プレーヤーの足下番号14），Tr.15（プレーヤーの足下番号15）のデータだけを，あとから作るもう一つのスタイルのアレンジデータにコピーして使いたいからです。Tr.12はピアノのバッキング音のトラックです。Tr.14はストリングス系（ヴァイオリン等の弦楽器）のトラックです。Tr.15はシンセサイザの打ち込み系のピコピコいうトラックです。打ち込み系ポップスにアレンジしたかったので，自分がいいなと思った打ち込みシンセトラックをポップスアレンジにプラスしたら，自分の想像どおりのアレンジになると思ったからです。

それから，つぎにもう一つのスタイルのアレンジをしました。またメロディを入力してスタイルパレットからJポップアイコンのスタイル選択画面からJ-Dance grooveを選択しました。そして，オートアレンジをする前に，今度はこの曲にイントロを付けたいので［編集］メニューから［すべて選択］を選び入力した音符データをマウスでドラッグ＆ドロップして，4小節分後ろに移動させました（**写真16**）。



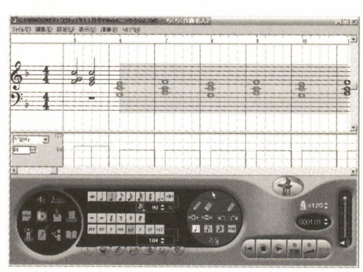
《写真16》編集メニューで音符データを編集

そしてオートアレンジの曲調確認画面で，今度は［明るく］を選択してみました。すると，先ほど移動させた4小節の部分にイントロを付けることができます。そのあとはコードをさっきのアレンジデータとまったく同じように差し替えて，セクションもFillを入れたりMainaA,Bを入れ替えたりしました。

ここまでできたら，またこのデータをスタイルパレットの中の楽譜に展開，パレットでアレンジデータを音符データにして保存しました。

さあここからがこの二つのアレンジを一つにあわせる作業です。まず，二つ目に作ったアレンジのTr.14とTr.15はシンセサイザのパッドと打ち込みのピコピコ音のトラックなので，このトラックに一つ目で作ったストリングスと打ち込みのトラックをもってきましょう。

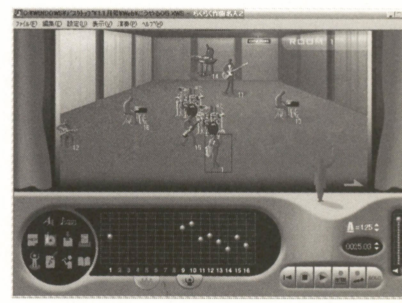
このTr.14，15にはすでに音が入っているので，それぞれの楽譜画面でイントロ4小節をはぶいて，それ以外音符の部分をマウスでドラッグして緑色の枠の中に入れて選択します（**写真17**）。そしたら［編集］メニューから［削除］をクリックして消します。この消した部分に前にアレンジしておいたトラックの音符データをコピー＆ペーストするのですが，まだピアノのトラックを入れるプレーヤーが足りないので，ステージ画面の指揮者の右にある右向きの矢印をクリックして，楽屋から1人，プレーヤーをステージに持ってきます。この状態でまた保存します。



《写真17》楽譜画面でイントロ4小節以外を緑色の枠の中に入れて選択

今度は一つ目に作ったアレンジを［ファイル］メニューから［開く］で読み込んで，足下番号12のプレーヤーの楽譜画面を開きます。［編集］メニューから［すべて選択］を選んで，また［編集］メニューから［コピー］をクリックします。そうしたら今度は，［ファイル］から二つ目に作ったアレンジのファイルを開いて，先ほど楽屋から持ってきたプレーヤーの6小節目からペーストしました。そしてこのデータを保存して，また一つ目にアレンジしたファイルを開き，Tr.14，15と順番にコピーしてはファイルを開いていきます。ストリングスだった音符データはパッドのプレーヤーに打ち込み，音はシンセサイザのプレーヤーにペーストしました。これで二つのアレンジデータのほしい部分だけを一つのアレンジとして合わせることができました。

あとは音色を変えたり，プレーヤーの位置を変えたり音のバランスを変えてできあがりです（**写真18**）。



《写真18》完成画面（上村）

完成したアレンジデータと，一つ目，二つ目に作ったアレンジデータはホームページからダウンロードして聴くことができるの参考にしてみてください。

◆Webの内容
- モチーフテンプレート
- 芸大生のモチーフアレンジデータ
- コンテストのくわしい応募方法

◆次回予告
- 「らくらく作曲名人2」作曲編
- 民族音楽のスタイル（コラム）
- シリーズ音響学（コラム）

「らくらく作曲名人2」作品募集！

らくらく作曲名人2で作ったアレンジデータを募集します。募集作品のテーマは，「今回の記事で取り上げたメロディをなんらかの目的のジャンルにアレンジしてみよう」というものです。

前回まではとにかくこのソフトに慣れるための練習みたいなものでしたが，今度は目的に合わせたアレンジということで，音楽性をいっぱい出してくれるとうれしいですね。

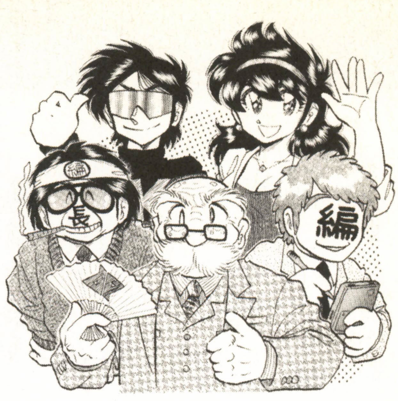
作品の発表会は来月のWeb上で行いますので，たくさんのご応募をお待ちしています。参加ご希望の方は，以下のデータを用意し，「らくらく作曲名人2・作品募集」係まで郵送か，またはEメールでお送りください。

・応募する際，用意するもの

①モチーフアレンジ・データ，MIDI・xwsファイル
②作品の簡単な紹介，コメント
③作曲者プロフィール
④連絡先（Eメール・アドレス）
（郵送の場合は，フロッピーディスクで）

http://www.basicmagazine.net
http://www.beeplus.or.jp/bm/art/

あくせすROOM



お便り連続回答

編：東京都練馬区・ルストフレムくんからのお便りに連続回答！

ル：この葉書が載るころには15歳です。

影：おめでとう。

つぐ美：おめでとう！

Dr.D：そして世界は平和になった。

ル：マイHPを持とうと決心。ツールなんぞ使わずにやっぱり『メモ帳』ですか。

編：ツールで作っても結局はソースをいじりますから，いずれはHTMLは学ぶ必要があります。

つぐ美：あとで面倒な思いをするか，先に苦労するか。

影：メモ帳は使いづらいから，せめてエディタぐらいは使ってもよろしい。

ル：この前，音楽で『カノン』のテストがありました。

影：サバです。

編：ニシンです。

つぐ美：タイ焼きよ。

ル：何かベーマガのためにできることはないですか？

影：それがわかれば苦労しない。

編：身近なネタでもいいので，投稿してください。

Dr.D：ベーマガに足りないと思うものを自分で作るのじゃ。

つぐ美：とりえず宣伝。

編：愛知県名古屋市・最馬場可くんの投稿に連続回答！

最：松原さんと従兄が似ている。彼女を連れてきて帰ってこいよ〜♥

影：きみのイトコは激辛砂糖水を売っていたのか？

つぐ美：デジタル露出計のプログラムができるなんてステキ。

Dr.D：もちろん「Mr.BOO!」の映画は全部見ているのじゃろう。

最：Catlikeのネコはカワイイですね。なにかグッズはありますか？

影：ゲームソフトが出ているぞ。

つぐ美：ご一緒に　クターもいかがですか。

Dr.D：疑問符のうしろには「。」をつけんでもいいぞ。1文字ぶん空けるのが原則じゃ。

うろおぼえ大募集

影：飲み屋でマンボウの刺身があってね。それで店の主人が「マンボウ描けるか？」と言い出したから，盛り上がってね〜。

編：マ，マンボウですか。背ビレとか胸ビレはありましたっけ？

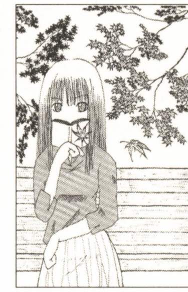
つぐ美：な，なに，この不気味な物体は!?

影：死んだような目つきってのは特徴をつかんでるね。

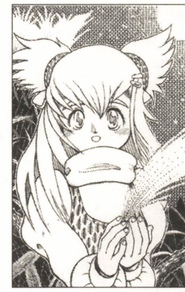
編：ああ，難しいですよ，資料がないってのは。

Dr.D：読者の日頃の観察力を試みる新企画「うろおぼえ友の会」では，毎月1問の課題を出すぞ。絵の巧拙を気にせずに，どんどん応募してくるのじゃ！

> うろおぼえ友の会
> 2002年1月号お題
> 「原動機付き自転車」
> （締切：2001年10月末日）
>
> ・資料や現物を見ないで描きましょう
> ・これがなんなのか人に聞かないでください
> ・うまく描けても賞品はありません



◀宮城県仙台市・吏久
編：残暑見舞いの予定が，秋も深まってしまったようです。



◀愛知県名古屋市・わいち301
つぐ美：モチーフはフクロウ？ そう言われてみると，なんとなく……。



▶愛知県豊田市・邪道坊
編：本を読みながら走るってのはどうですか？ 「走れメロス」とか。



▶千葉市・バグニンゲン
影：別冊化しろという圧力だろうか。
編：不思議議と人気コーナーでした。

素材名人 萌えプロ

影：やたら反応の多かった今月のお題は「メイドさん」「楽器が好き」。欲を言えば，もっと斬新な衣装デザインの投稿がほしかったが，読者も，描き手も，癒し系のイラストを求めていたようなので，今回はうるさく言わないことにしよう。こういうテーマが出題されたら，とにかく図書館や書店やビデオショップへ行って資料を探そう。みんなが思い浮かぶパターンは，あくまで世の中で広く流通しているデザインというだけで，本当に各人がベストだと思っているデザインだとはかぎらないのだから。



◀福島市・南斗ライバ
編：曲名は「萌えろ荒鷲」だそうです。
影：地下室で秘密練習中なのかな。



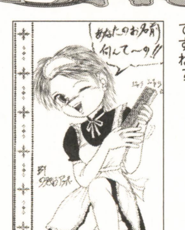
◀愛知県刈谷市・せゆ
つぐ美：お仕事中でも音楽を楽しんでいます。



◀大阪府松原市・楸様江
影：うしろの3人がえらくアヤしい……。いや，3匹か。



◀千葉県柏市・鈴木有人（故障中）
影：葉書の裏に楽しい人物設定が。シリーズ化希望〜。



◀愛知県名古屋市・のうぎょうアッパー
編：えーと，これはラップの一種ですね？



◀福岡県糟屋郡・MINO
影：にぎやかで楽しそうだけど，毎度聞かされるほうは，もうウンザリ？



◀愛知県丹羽郡・裁刃鋭人
つぐ美：アコーディオンの重さも顔には出さず，スマイル，スマイル！

お題と締切日

影：今回はフリルを描く練習になったかな？　次回でサヨナラの「萌えプロ」では，障害物競走っぽいネタが多そうなので，あらかじめ釘をさしておこう。爆炎から脱出するようなハリウッドばりのスタント・シーンは，描くのが難しいぶん，応募が少なさそうだ。

11月号
・メイドさん
・楽器が好き
・（自由課題）
（今月号で公開）

12月号
・防災訓練
・菓子パン
（締切：2001年10月15日）

あくせすROOM応募のルール

●愛読者カードや，官製ハガキ，同サイズの丈夫な紙などでお送りください。

ワンポイント
まとめ送りで大量に送るよりも，気合いの入った数枚の作品のほうが，インパクトは強く，掲載率は上がるようです。

●鉛筆や色鉛筆，涙滴の多いボールペン画は印刷に不向きなので避けましょう。

ワンポイント
下書きのあとが消し切れてないと見苦しくなり，評価が大幅にダウンしてしまいます。仕上げはていねいに。

●投稿内容は，掲載されたプログラムの感想や，学校でのできごと，一発ギャグなど，なんでも結構です。みなさんで楽しいページにしていきましょう。

≪応募先≫
〒141-8715 東京都品川区東五反田1-11-15
電波新聞社マイコンBASICマガジン編集部
あくせすROOM係

webでも受付中です。Internet読者ハガキでどうぞ。
http://www.basicmagazine.net/

Falcom® NEWS EXPRESS

日本ファルコム株式会社
〒190-0012 東京都立川市曙町1-14-13
TEL 042-527-6501
FAX 042-528-2714

スタッフ募集中
採用担当BFN係

http://www.falcom.co.jp/

PCゲーム定番キャンペーン スタート! (9/29～11/30)

マウスシート全員プレゼント!

いよいよスタートしたファルコムPCゲーム定番キャンペーン！「イース」シリーズや「英雄伝説ガガーブトリロジー」シリーズをはじめ，国産RPGを代表する定番ソフトを，まだプレイしていないタイトルがあれば，この機会にぜひプレイしよう！キャンペーン実施店は，ファルコムホームページのショップリストに掲載中。★マークが付いている店でソフトを購入すると，軽くて薄い超便利なマウスシートがもらえるぞ！期間中でもなくなったら終了となるのでお早めに！

Http://www.falcom.co.jp./shoplist/index.html

「マウスシートプレゼント対象ソフト」

- ■イースⅠ・Ⅱ完全版
- ■英雄伝説Ⅳ朱紅い雫
- ■ソーサリアンオリジナル
- ■イースⅡエターナル
- ■新英雄伝説Ⅲ白き魔女
- ■英雄伝説Ⅴ海の檻歌
- ■西風の狂詩曲
- ■ファルコムゲームBOX Vol.1

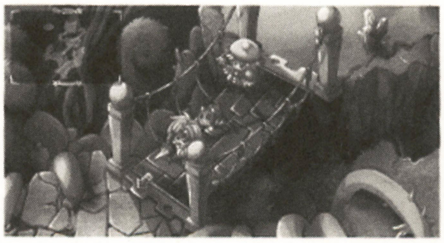


▲全8種の中から，ソフト購入1本につき1枚プレゼント！

完全新作「ツヴァイ!!」12月発売決定！



遂にファルコム完全新作「ツヴァイ!!」の発売が12月に決定！イースやザナドゥ，ブランディッシュに代表されるように，アクションRPGといえばファルコムのお家芸。今回の「ツヴァイ!!」も，ファルコムテイスト満載の，軽快なアクションが楽しめるぞ！「ツヴァイ!!」では主人公のピピロとポックルの２人を，状況に応じて交代させながら冒険を進めていくのだ。冒険するステージを自由に選べたり，２人で協力して必殺技を出すなど新たな試みも満載だ！新シリーズ「ツヴァイ!!」に乞うご期待！



イースⅠ・Ⅱ 限定1000セット 2002年カレンダーマウスシート好評発売中!

キャンペーンでもらえるマウスシートとは別デザインの，イースⅠ&Ⅱのイラストを使った2002年カレンダー付きマウスシート（４枚組）をただ今，ファルコムホームページのオンラインショッピングコーナーで好評発売中！1000セットの限定販売なので，お申し込みはお早めに！



ソーサリアンforever ベクター販売ランキング1位をGet!

ベクタープロレジで好評発売中の「ソーサリアンforever」が，ゲームではもちろん，ウイルス駆除のユーティリティやインターネット関連ソフトを含めたすべてのジャンルの中で販売数ランキング1位を獲得！ベクターで購入された方には「ソーサリアンMIDIコレクション」も特典に付いているので，まだソーサリアンをプレイしたことがないという方は，お得なプロレジ販売を利用しよう。

◆購入を希望される方はこちらにアクセス！　価格：1980円
http://www.vector.co.jp/swreg/catalogue/sr025409/?srno=SR025409&site=v
ベクターで販売中のファルコム製品が一目でわかるこちらもご覧ください。
Http://www.falcom.co.jp./vector/index.html

東京ゲームショウでイースⅠ&Ⅱのグッズを限定発売!

10/12～14に開催される東京ゲームショウでイースⅠ・Ⅱのテレカとトレーディングカードの発売が決定！ホビージャパンのブースで発売されるので，ゲームショウに行かれる方は，確実にゲットしよう！ファルコムオンラインショッピングでも発売予定。

●YSⅠ・Ⅱトレーディングカード・全54種
価格4800円（1BOX 8パック入り・1パック6枚）
＊トレカはBOX発売のみ
●YsⅠ・Ⅱテレホンカードセット A・B（各3枚）
価格：各4500円



12月に発売が決まった「ツヴァイ!!」はどのシリーズにも属していないファルコムの完全新作タイトルです。シリーズ続編の新作はけっこう出ていますが，新シリーズは久しぶりですのでスタッフ一同はりきって，いろんなアイデアを盛り込んでいるようです。発売まであと少しですので，皆さんお楽しみに！

SEGAのヘンな話題やちょっとお得な情報ならおまかせ!!

ベーセガ

マイコン BASIC
SEGA OFFICIAL
VARIETY SPACE

奈美 みなさーん，こんにちは！　今私は東京ゲームショウ会場のセガブースに来ています。今月の「ベーセガ」は，幕張メッセより中継でお送りしたいと思います～！

竹崎 えっ!?　ゲームショウは10月12日からだよー。今日は8日だし，まだ始まってないんじゃない？　ここはどこ？　未来，未来？

奈美 いえ，気にしないでください。おおお，ブース内にはたくさんの人がぁー，人がぁー。み，見える，みえる～！

竹崎 見える，みえる，あなたの未来が見えるぞよ～！って，さては奈美さん，いよいよ幻覚を見るようになってしまったのかー。

奈美 もー，違いますよー。ちょっとゲームショウの雰囲気をかもし出してみようと思っただけなのにー。ちぇっ。

竹崎 「ちぇっ」って…。そーゆーことは，ちゃんとリハーサルのときに言ってくれないと，わかんないじゃん！

奈美 えええーい，黙らっしゃーい！　てな訳でみなさまご存知の業界最大のゲームイベント・東京ゲームショウにセガが帰ってきまーす！

竹崎 なんと，昨年春に出展して以来，3回ぶりの復活だぁー！　しかもスゴイぞ！

奈美 そこで，今月の「ベーセガ」は，ゲームショウ前に公開された情報をみなさんにお知らせしちゃいます。

竹崎 では，早速行くぜー。「サクラ3」の人気も冷めやらぬ中「サクラ4」の電撃発表や「サクラ大戦GB2」発売決定など，ますます広がりを見せる「サクラ大戦」プロジェクト！

奈美 12月公開予定の映画「サクラ大戦活動写真」に合わせるかのように，シリーズ初のオンラインゲームがDCに登場します！

竹崎 「サクラ大戦オンライン」は，おなじみのキャラたちや各プレイヤー同士でゲームの対戦をしたりコミュニケーションをとったりといったソフトなんだ。

奈美 パートナーとなるサクラキャラ達は，本編と同じように，会話の内容次第で機嫌やセリフが変化しますよ。

竹崎 ゲーム対戦にはパートナーキャラも参加しているから，こりゃ負けられないぞ。

奈美 はい。チャットにゲームにデートにと大忙しですね！

竹崎 そして，ファン待望の「サカつく特大号2」の発売日も12月に決定だ！　これはこの前の「やきゅつく&あそ」同様，ファンの方からの熱いリクエストで開発決定となったソフト。しっかり買って&遊んでほしいよね～！

奈美 「やきゅつく」といえば，ついにこの人気シリーズがGBA用でも遊べるようになるんですよー。

竹崎 おお，これで電車での移動中だって，車の運転中だって……（バキ！）

奈美 運転中はできませんッ！！

竹崎 にしても，これでいつでもどこでも「やきゅつく」ができるのだ！　その名も「どこでもやきゅつく」！

奈美 そのまんまじゃないですか！

竹崎 わかりやすくていいじゃん。

奈美 さらに，どこでもといえば「ソニックアドバンス」！

竹崎 「チャオガーデンアドバンス（仮）」機能搭載で，GBAでチャオが育成できるようになったのだ！

奈美 これで，ちょっとした空き時間にもチャオをせっせと育てられますねっ。

竹崎 なんとGBAで育てたチャオをキューブ版「ソニック　アドベンチャー2」へ送ってチャオレースに出すこともできる！

奈美 GBAオリジナルのチャオもいるらしいですよっ。これは目が離せませんねえ。

竹崎 GBAといえば，あの「ザ・ハウス・オブ・ザ・デッド」がピンボールゲームになっちゃいました。

奈美 ハイハイ「ザ・ピンボール・オブ・ザ・デッド」のことですね？

竹崎 という具合に今後もますますセガから目が離せないぜ～！

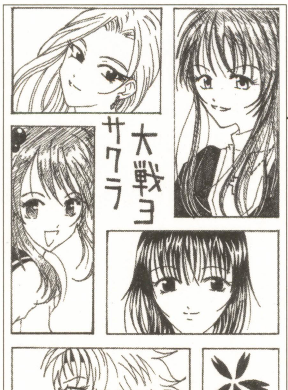
奈美 ドリームキャストから他機種ソフトまで内容も盛りだくさん！　みなさん，ぜひセガブースに遊びにきてね～！

竹崎 東京ゲームショウは，幕張メッセにて，10月12日～14日の3日間開催されているぞ！



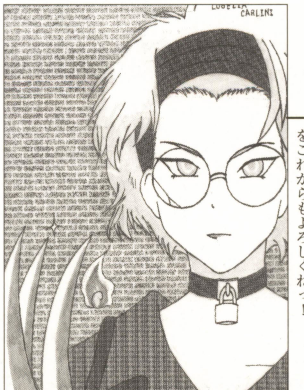
◀ぱんだぱぱ

PSOのPC版も発表になりましたね！　ゲームキューブ版といい，まだまだ眠れない夜が続きそうです！



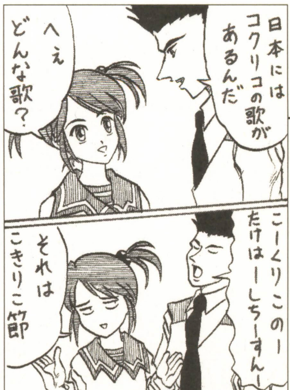
◀葉月魁斗

いつもよりちょっとだけ大人（!?）のエリカたち。ありがとうございます。最近は涼しくなりましたね！



◀吏久

シザーハンズを構えるロベリア。「ベーセガ」をこれからもよろしくねっ！



◀大山修史

絶妙のツッコミですね。いいコンビです。個人的には，その2も面白かったです（笑）

このコーナーを充実させるための，ご意見，お便り，イラストなどなどクビを長くしてお待ちしています。採用の方には特製テレカなど，非売品セガグッズをプレゼント。SEGA竹崎がお便りをすべて読んで，面白いコーナーを作っていきたいと思います。すべてのあてさきは，ここや!!

〒144-8531 東京都大田区羽田1-2-12 （株）セガ カスタマーゲートウェイ部「竹崎と奈美のベーセガ」係

セガ竹崎の「セガスタ」はこちら→http://www.sega.co.jp/segasuta/



ニンテンドーCube対応？

なにわ通信

PRESENTED BY マイコンソフト

ADDRESS● 〒530-0005大阪市北区中之島3-2-4
朝日新聞ビル6F 電波新聞社大阪本社内
マイコンソフト株式会社 なにわ通信係
http://www.micomsoft.co.jp/

タイトル冠文：大阪府・大阪難波
タイトル・イラスト：千葉県・鈴木有人
本文：暁

アメリカで痛ましい事件が起きてしまいましたね。(この原稿は，事件のあった次の日9月12日に書いています)

テロだそうですが，民間の旅客機でビルに突っ込むなど，とても信じられません。

XRGB-2plus

さて，いよいよ新製品「XRGB-2plus」の発売が迫ってきました。

XRGB-2plusは，ゲーム機の映像をパソコン専用ディスプレイに表示する装置で，ゲーム機の映像とパソコンの映像を切り替えて表示できる，たいへん便利な装置です。

前号では，新製品の機能比較表を掲載しましたが，発売も近づいておりますので，懲りもせず，再度ご紹介させていただきます。

今回は一歩踏み込み，代表的な機能についてご説明したいと思います。

ディスプレイとテレビの違い

まず，パソコン用ディスプレイと家庭用テレビとの違いについてご説明します。

パソコン用ディスプレイと，家庭用テレビでは，映像端子の形状や同期周波数が違うだけでなく，高繊細なブラウン管を使用することによって発生する問題がいくつかあります。

具体的な例では，家庭用テレビに比べ画面が全体的に白っぽいとか，家庭用テレビの赤色とディスプレイの赤は違う，などです。

また，ディスプレイは，仕様上の制限などから，家庭用テレビに比べ画面が暗いです。

これらの問題があるため，パソコン用ディスプレイで家庭用テレビと同じ色合いや，同じ明るさを再現することは，ほぼ不可能です。

こういった差異をできるだけ抑えるため，XRGB-2plusには下記のような機能が用意されています。

＜黒レベル調整機能＞

同期信号のレベルを変化させるなどの処理を行うことで，画面の「黒さ」を調整することができます。この機能を使用すると，映像の黒い部分を強調することができるので，映像の色が強調され奇麗に見えます。

＜ガンマ補正機能＞

パソコン用ディスプレイは明るさの最大値が決められており，その規格を越えてしまうと，画面に色ダレなどが発生してしまいます。

このため，画面の黒い部分(最も暗い部分)と白い部分(最も明るい部分)を変えることなく，画面を明るく見せる技術が必要となります。その技術がガンマ補正です。

XRGB-2plusには，通常のガンマ補正の他に，R，G，B個別にガンマ調整できる機能も付いており，微妙な色合いも調整できます。

＜シャープ機能＞

XRGB-2plusには，独自の画像処理によって4種類の鮮明度が用意されています。

この機能により，S映像ではエッジが強調された鮮明な映像表示が可能となりました。

このシャープ機能はD1入力やRGB入力でも使用できます。

発売予定

XRGB-2plusの発売予定は，11月上旬です。予定価格は28,800円(税別)です。

お知らせ

前号では，XRGB-2plusのガンマ補正機能は「96段階」調整可能であるとお伝えしましたが，仕様の変更により「79段階」となりました。ご了承ください。

ありがたいおハガキのコーナー



大阪府・藤井朱丸
藤井さんは，冒険小説にはまっておられるそうです。司馬遼太郎，船戸与一，佐々木譲がオススメとのこと。



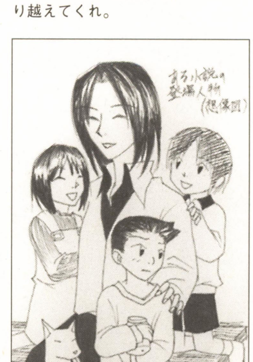
大阪府・ARAI
熊恋シリーズも佳境。立てっ！立ち上がって過去に立ち向かうんだっ！全てを受け入れ，それを乗り越えてくれ。



愛知県・邪道坊や
宇宙戦艦の機関室の光景です。厳しそうな機関長ですが，普段はきっと優しい「おやっさん」なのでしょうね。



千葉県・バグニンゲン
このハガキを手元に置くのが恥ずかしいそうです。XRGB-2星人につきまとわれるサラリーマンのお話。…こんな人生イヤだ〜っ！



福島県・冬虫夏草
ある小説の登場人物。とのことですが，いったいどんな小説なのでしょうか。真ん中の人はお母さんでしょうか。



大阪府・まぢけん様江(参考図)
0.05のピグマで描かれたそうです。手軽な画材で絵は描けます。
読者の皆さんもイラストに挑戦してみましょう！

毎年生徒数増加！就職最先端校！

今年も求人数増加！すでに就職内定者が続出

進路指導部が中心となり、学院に寄せられる求人票に従って、自作を持参し面接を受ける。こうして次々に就職内定者が決まっています。

初心者が卒業時にはゲームデザイナーで就職

初心者でも自分の好きなことなら、目標達成までがんばれます。2年間の在学中に、現場をよく知っている経験豊かな先生のきめ細かな指導で、就職するための技術を習得できます。

さらに学費軽減！入学時費用56万円でOK!!

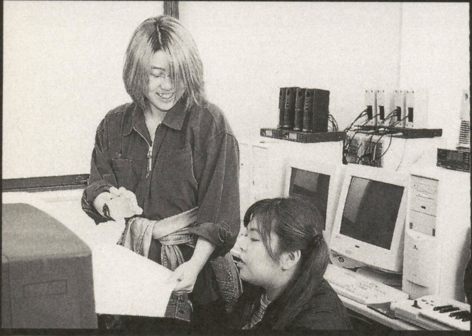
学院では平成14年度入学生から、学費を大幅に値下げしました。今まで108万円の学費を84万円に、2年次の学費も90万円を66万円にしました。入学時の費用も56万円となります。

毎月6276円で学べる教育ローンが大評判

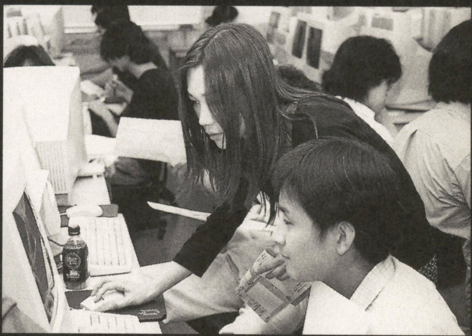
本学院が提携する「三井住友銀行教育ローン」を利用すると、毎月6276円で学べます。この教育ローンは ①入学生の保護者を対象にし ②融資金額も500万円まで ③融資期間7～14年と有利です。(例・80万円の学費借入の場合、14年間返済で毎月6276円)借入れも容易なので毎年利用者も急増しています。

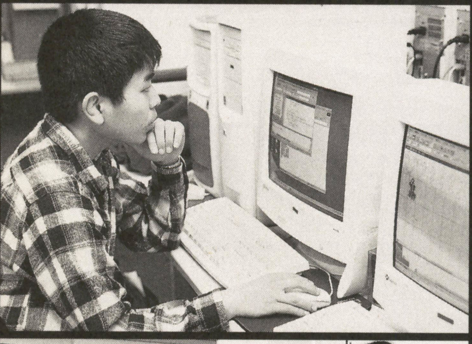
見逃せないぞ！これが14年度入学生の特典だ

1. 180万円を84万円に！年間学費の軽減実施！
2. 学院奨学金制度。入学生に3万～12万円支給
3. 生徒数増加で設備拡充。最新マシンで実習
4. 就職先企業の採用担当者出席の就職説明会
5. 業界に精通している講師の拡充と担任制度
6. 自分が得意な学科を選べる選択授業の充実
7. 自主制作のため教室は午後7時頃まで開放





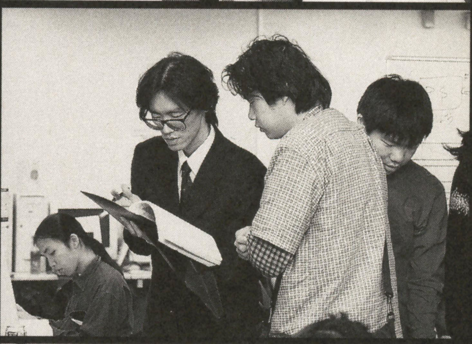




14年4月生願書受付中

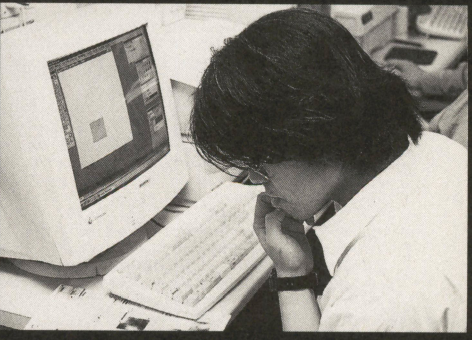
無料体験実習講座・学院見学！

体験実習講座は10月13日、10月27日の土曜日午後1時。学院見学・入学相談は毎日実施しています。参加特典もあり。電話予約し来校してね。







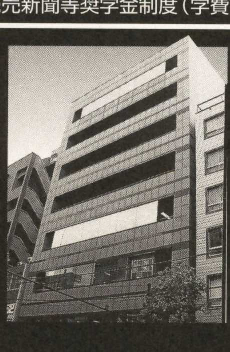


■就職抜群！学生寮完備。朝日・毎日・読売新聞等奨学金制度（学費全額免除・給与月額12万円支給）三井住友銀行教育ローン有。学院見学随時可⇨日時をTELにて要予約

14年4月生願書受付中

ゲームデザイナー学部
全日制2年・定員300名

ゲームデザイン科
プログラム科
グラフィック科
キャラクターデザイン科
デジタルサウンド科



文部省認可CGARTS協会認定CG教育校／社団法人・日本漫画家協会賛助会員／社団法人・日本映画テレビ技術協会会員

日本ゲームソフト学院

〒101-0061 東京都千代田区三崎町3-4-9 ☎03-3261-8211

入学案内書 はがき（住所・氏名・年令・電話番号明記）か、電話で上記まで（無料）

FAX 0120-194-750 http://www.tag.co.jp/ E-mail:ngs@tag.co.jp

中古通販

PC NET ピーシーネット

在庫保有数日本一
(常時10,000台)

TEL.03-3733-1441

通販営業時間／月～金 9:30～19:00 土・日・祝日 11:00～19:30

www.prins.co.jp/

楽天市場店 http://www.rakuten.co.jp/pcnet/ ネットプライスモール http://www.netpricemall.com/
ジャストネット http://email.justnet.ne.jp/shop/pcnet/ ビッグローブショッピングワールド http://shopping.biglobe.ne.jp/

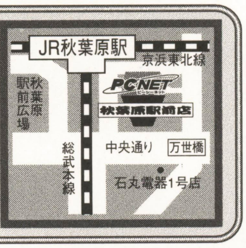
最新情報は電話かネットでチェック!! 急げ!!

10月25日(木)11:00より OPEN

秋葉原駅前店

駅を出てスグ! 徒歩1分

イベント多数有り



インターネットでは毎回お得な企画がいっぱい!!

3日間保証セール だから安い!!

期間限定タイムセール

[安心の全品保証付] 1万円以上…1ヵ月保証 1万円以下…1週間保証

※本商品は添付品(OSを含む)は付属しておりません。
※クリーニングご希望の方、別途¥1,000が必要です。
※中古パソコン販売時の在庫定義が変わりました。
今後在庫とは、従来品に加え、動作上支障のないキズ・天板やけ・カバー等の欠品がある物や、キーボード等が補充(他メーカー含)されている商品も、「在庫」としてお取り扱いさせていただきます。

DOS/V

注文No.	商品	価格
U1772	CelebrisFX-2 5200M(DEC)(MP(200)/32MB/2.1GHD/CD12/外装焼け/外装塗装剥げ/PCIX1/ISAX1/共有X1)	¥11,800
U8713	Aptiva(2188-47J)(IBM)(Celeron(466)/128MB/8.4GHD/CD24/スピーカ・ジョイスティック無し/省スペース型)	¥37,800
U8575	PC300GL(6282-70J)(IBM)(Pen(200)/96MB/2.5GHD/CD16/PCIX1/ISAX1/共有X2/USBX2)	¥15,800
U2732	PC350(6587-JBV)(IBM)(MP(200)/64MB/2.5GHD/CD8/ISAX2/共有X3)……	¥9,800
U8684	PC350(6587-JC3)(IBM)(MP(200)/128MB/4.2GHD・3.2GHD(HDD2つ付き仕様)/CD16/ISAX2/共有X3)	¥19,800
U8714	PC300PL(6562-50J)(IBM)(MP(233)/98MB/4.2GHD/CD24/PCIX2/ISAX1/共有X1/USBX2)	¥18,800
U8538	PC300GL(6561-27J)(IBM)(Celeron(266)/32MB/4.2GHD/CD32/外装焼け/PCIX2/ISAX1/共有X1/USBX2)	¥18,000
U8708	PC300GL(6561-55,56J)(IBM)(PenII(300)/64MB/4.2GHD/CD32/汚れ/PCIX2/ISAX1/共有X1/USBX2)	¥29,800
U1596	PC300PL(6871-12J)(IBM)(Celeron(400)/64MB/8GHD/CD40/省スペース型)‥	¥32,800
U8706	PC-MA26DS7BAA41(NEC)(PenII(266)/32MB/4.3GHD/CD24/外装焼け/PCIX4/USBX2)	¥16,800
B0182	PC-MA45DSZAMA63(NEC)(PenII(450)/32MB/6.4GHD/CD32/100TX)……	¥35,800
U1639	PC-MA23DCHBAB31(NEC)(PenII(233)/64MB/3.2GHD/CD20/擦り傷)……	¥19,800
U1660	DESKPRO 4000/6266(COMPAQ)(PenII(266)/96MB/3.2GHD/CD24/タワー型/PCIX4/ISAX3)	¥22,800
U1790	DESKPRO EP C266/3.2(COMPAQ)(Celeron(266)/32MB/3.2GHD/CD/擦り傷/PCIX3/ISAX3/USBX2)	¥17,800
V0157	DEPEP 6400C6.4CDS98(COMPAQ)(Celeron(400)/128MB/6.4GHD/CD32/100TX/Win98/付属品付)	¥32,800
U8443	PROSIGNIA 200(COMPAQ)(PenII(233)/128MB/4.3GHD/CD16/フルタワー型/PCIX3/ISAX1/共有X1)	¥19,800
U1264	OPTIPLEX GXI(MP-200)(DELL)(MP(200)/32MB/3GHD/CD8/外装焼け/キーボード無し/高さ10.9cm)	¥7,800
U8426	OPTIPLEX GXI(DELL)(MP(200)/32MB/2.1GHD/CD8/外装焼け/汚れ/高さ16.5cm/PCIX2/ISAX2/共有X1/USBX2)	¥15,800
U8696	OPTIPLEX GXI(DELL)(MP(233)/130MB/4.2GHD/CD/ミドルタワー型)………	¥17,800
U8532	OPTIPLEX GX1(DELL)(Celeron(333)/64MB/4.3GHD/CD/外装焼け/横置き型/PCIX1/ISAX1/USBX2)	¥27,800
U8701	Dimension XPS R400(DELL)(PenII(400)/128MB/6GHD/CD/リアカバー無し/外装焼け/ミドルタワー型)	¥37,800
U8571	GATEWAY 2000P5-200(GATEWAY)(MP(200)/32MB/2.5GHD/CD4/目立つ外装焼け/ミドルタワー型/PCIX3/ISAX2/共有X1)	¥10,800
U8685	GATEWAY 2000(GP6-266)(GATEWAY)(PenII(266)/64MB/3.2GHD/CD/外装焼け/47cm)	¥20,800
U8584	GATEWAY 2000(G6-266)(GATEWAY)(PenII(266)/128MB/4.2GHD/CD/外装ヤケ/擦り傷/横置き型(14cm)/PCIX3/ISAX2/共有X1/USBX2)	¥24,800
V0144	ZHP-D6582N/801(HP)(PenII(400)/128MB/6.4GHD/CD32/100BT)…………	¥34,800
U8709	FMV-9D50A11.61(富士通)(MP(233)/64MB/3.2GHD/CD/外装焼け/USBX2)‥	¥18,800
U1413	FMV-626T6C5(富士通)(PenII(266)/32MB/4.3GHD/CD16/外装キズ/タワー型)	¥24,800
U1714	FMVT42671.42672.42673共通(富士通)(PenII(266)/64MB/4.2GHD/CD24/PCIX3/ISAX1/共有X1/USBX2)	¥21,800
U1923	FMVT53071.2.3(TV307)(富士通)(PenII(300)/96MB/4.3GHD/CD32/スピーカ無し/モデム/外装焼け/擦り傷/PCIX3/ISAX1/共有X1/USBX2/AGPX1)	¥26,300
U1071	FMV7T90B11.61(FMV6300T7)(富士通)(PenII(300)/32MB/4.3GHD/CD20/タワー型/PCIX3/ISAX1/共有X1/USBX2/AGPX1)	¥22,800
U8717	FMV-5SLM121,101,161(FMV-6566SL5C)(富士通)(Celeron(566)/64MB/10.2GHD/CD40/外装塗装剥げ/台座無し/縦・横共通/PCIX3/USBX2)	¥49,800

DOS/Vノート

注文No.	商品	価格
U3846	ThinkPad 315D(2600-70J)(IBM)(Pen(133)/32MB/1.08GHD/CD10/SVGA/11.3TFT)	¥32,800
U6712	ThinkPad 365X(2625-2J9)(IBM)(Pen(133)/24MB/1.08GHD/SVGA/11.3DSTN)	¥15,800
U6377	ThinkPad 365X(2625-1J9)(IBM)(Pen(133)/24MB/1.35GHD/CD6/SVGA/11.3TFT/16MB増設)	¥37,800
U6554	ThinkPad 380D(2635-3AJ)(IBM)(MP(150)/48MB/2.1GHD/CD8/SVGA/12.1TFT)	¥39,800
U6852	ThinkPad 380E(2635-40J/49J)(IBM)(MP(150)/32MB/2.1GHD/SVGA/リアカバー無し/12.1DSTN)	¥19,800
U6505	ThinkPad 535(2606-MF9)(IBM)(Pen(133)/40MB/4.6GHD/SVGA/外付けFDD/B5サイズ/外装焼け/10.4TFT)	¥27,800
U3354	ThinkPad 535E(2606-36J)(IBM)(MP(150)/48MB/1.6GHD/SVGA/10.4TFT)‥	¥37,600
U3694	ThinkPad 560E(2640-40J)(IBM)(MP(166)/16MB/2.1GHD/SVGA/12.1TFT)‥	¥29,800
U6558	ThinkPad 560X(2640-65J)(IBM)(MP(200)/64MB/2.1GHD/SVGA/擦り傷/12.1TFT)	¥40,800
U6888	ThinkPad 600E(2645-8AJ)(IBM)(PenII(366)/160MB/6.4GHD/CD24/XGA/13.3TFT)	¥76,800
U6544	PC-VA23C/WXA(NEC)(MP(233)/64MB/2.1GHD/CD20/XGA/外装塗装剥げ/13.3TFT)	¥54,800
U6117	LTE5400(COMPAQ)(Pen(150)/16MB/2.16GHD/CD6/SVGA/12.1TFT)……	¥32,800
U6774	ARMADA 1550DMT(COMPAQ)(Pen(133)/16MB/1.4GHD/CD10/SVGA/PCカバー無し/12.1TFT)	¥26,800
U6864	ARMADA 1580DMT(COMPAQ)(MP(150)/82MB/2.1GHD/CD10/SVGA/モデム/12.1TFT)	¥39,800
U6863	ARMADA 1590DMT(COMPAQ)(MP(166)/82MB/2.1GHD/CD/SVGA/キーボード文字剥げ/12.1TFT)	¥46,800
U6274	FMV-5120NU2/W(FMV-52NU2W5.1共通)(富士通)(Pen(120)/32MB/1GHD/SVGA/外装キズ/外装焼け/11.3TFT)	¥13,800
U6083	FMV-NU13D(富士通)(Pen(133)/32MB/1.32GHD/CD20/擦り傷/12.1DSTN)…	¥24,800
U6100	FMV-NU5163(富士通)(MP(166)/32MB/2.1GHD/CD14/SVGA/擦り傷/12.1TFT)	¥45,800
U6855	FMV-1NU2WA5★添付品付き(富士通)(MP(233)/48MB/3.2GHD/CD/SVGA/Win95付/マニュアル付/12.1TFT/USBX1)	¥56,800
U6854	FMV-1NU2WA5★添付品付き(富士通)(MP(233)/48MB/3.2GHD/CD/SVGA/Win95付/Win98アップグレード付/マニュアル付/12.1TFT/USBX1)	¥59,800
U6845	FMV-1NA2XCO.6.5共通(富士通)(MP(233)/96MB/4.0GHD/CD24/XGA/外装キズ/13.3TFT/USBX2)	¥59,800
U6873	FMV-2NU6X0,6,5(富士通)(PenII(233)/32MB/3.2GHD/XGA/キーボード文字剥げ/13.3TFT/USBX1)	¥39,800
U6875	FMV-6266NU3/X(FMV-3NU7XA0.5.6)(富士通)(PenII(266)/32MB/4.3GHD/XGA/CD付き/13.3TFT/USBX1)	¥62,800
U6840	FMV-NE560C1,2,3(富士通)(Celeron(600)/64MB/15GHD/CD24/XGA/14.1TFT)	¥116,000
U3326	DynaBook Satellite220(東芝)(Pen(133)/16MB/1.4GHD/SVGA/外装塗装剥げ/12.1DSTN/PA1240S9)	¥13,800
U6735	PA1262C9(東芝)(MP(200)/64MB/2.1GHD/SVGA/CD無し/PCカバー無し/外装塗装剥げ/12.1TFT/USBX1)	¥32,800
U3789	AL-N2T515J5(パナソニック)(MP(150)/32MB/1.6GHD/SVGA/外装塗装剥げ有り/10.4TFT)	¥39,800
U6769	GATEWAY2000 SOLO2500(GATEWAY)(MP(233)/64MB/2.1GHD/DVD/SVGA/擦り傷/リアカバー無し/12.3TFT)	¥69,800

バッテリーパックは消耗品につき保証対象外とさせていただきます。

98デスクトップ

注文No.	商品	価格
U2627	PC-9821V200/SZC2(MP(200)/32MB/3GHD/CD20/スピーカ無し/外装焼け/外装塗装剥げ)	¥28,800
U2632	PC-9821V233/M2(MP(233)/32MB/3GB/CD20/Fax36.6K/ミニタワー/外装焼け/外装キズ/スピーカ・マイク無し)	¥42,800
U8722	PC-9821V233/M2(MP(233)/64MB/3GB/CD20/モデム/スピーカ・マイク無し/擦り傷/ミドルタワー型)	¥45,800
U1636	PC-9821XC16/S5A3,S5B3(Pen(166)/32MB/1.6GHD/CD8/汚れ/擦り傷)…	¥14,800
U8610	PC-9821RA223/N30(PenII(233)/64MB/3GHD/CD16/LAN/外装ヤケ/PCIX2)	¥35,800
U8585	PC-9821RA223/W30(PenII(233)/96MB/3GHD/CD16/外装焼け/外装キズ/PCIX2)	¥38,800

98ノート

注文No.	商品	価格
U3057	PC-9821NM/340(486DX2(50)/1.6MB/340HD/擦り傷/モノクロ)…………	¥14,800
U3059	PC-9821ND(486SX(33)/1.6MB/外装焼け/9.5DSTN)…………………	¥12,800
U0345	PC-9821NE3/3(486DX2(50)/8MB/350HD/外装キズ/外装焼け/10.1TFT)…	¥14,800
U6273	PC-9821NE3/3(486DX2(50)/16MB/350HD/VGA/外装塗装剥げ/10.1TFT)	¥13,500
U3334	PC-9821NF/810(Pen(75)/7.6MB/810HD/PCM/擦り傷/9.5TFT)…………	¥13,800
U3187	PC-9821NP/340(486DX4(75)/5.6MB/340HD/擦り傷/9.5TFT)…………	¥12,800
U2256	PC-9821NP/540(486DX4(75)/21.6MB/540HD/外装焼け/9.5TFT/16MB増設)	¥15,800
U3135	PC-9821LT2/7A(486DX2(50)/8MB/720HD/サブノート/外装塗装剥げ/9.5TFT)	¥15,800
U3311	PC-9821LA10/S8C(Pen(100)/40MB/810HD/サブノート/24MB増設済み/PCカバー無し/10.4TFT)	¥14,800
U6829	PC-9821LA13/S14(Pen(133)/48MB/1.4GHD/VGA/サブノート/10.4TFT)…	¥29,800
U3269	PC-9821NB7/5(Pen(75)/8MB/540HD/PCM/擦り傷/10.4TFT)…………	¥9,800
U6594	PC-9821NA9/H8(Pen(90)/48MB/810HD/PCM/擦り傷/CD付き/10.4TFT)…	¥19,800
U6275	PC-9821NA12/S8(Pen(120)/32MB/810HD/CD4/PCM/12.1TFT)………	¥29,800
U6699	PC-9821NA12/H10(Pen(120)/80MB/1GHD/CD4/PCM/SVGA/擦り傷/10.4TFT)	¥31,800
U3075	PC-9821NA13/H10(Pen(133)/16MB/1GHD/CD4/PCM/SVGA/12.1TFT)‥	¥31,800
U3312	PC-9821NA13/C10(Pen(133)/16MB/1GHD/CD4/PCM/VGA/10.4TFT)…	¥27,800
U0326	PC-9821NR13/D10B(Pen(133)/32MB/1GHD/CD8/PCM/12.1DSTN)……	¥29,800
U6490	PC-9821NR13/S14(Pen(133)/32MB/1.4GHD/SVGA/外装塗装剥げ/12.1TFT)	¥22,800
U6393	PC-9821NR15/S10(Pen(150)/24MB/1GHD/CD8/外装塗装剥げ/12.1TFT/8MB増設)	¥33,800
U6331	PC-9821NR15/S10(Pen(150)/32MB/1GHD/CD8/SVGA/12.1TFT)………	¥36,800
U6441	PC-9821NR15/S10(Pen(150)/80MB/1GHD/CD8/SVGA/12.1TFT/64MB増設)	¥37,800
U3852	PC-9821NR150/X14F(MP(150)/32MB/1.4GHD/CD10/モデム/XGA/リアカバー無し/12.1TFT)	¥49,800
U6806	PC-9821NR233/S32A(MP(233)/32MB/3.2GHD/CD16/SVGA/外装擦り傷/12.1TFT)	¥68,800
U1647	PC-9821NW133/D14C.D共通(MP(133)/32MB/1.4GHD/CD11/SVGA/12.1DSTN)	¥24,800
U6075	PC-9821NW150/S20C(MP(150)/32MB/2GHD/CD16/モデム/SVGA/12.1TFT)	¥54,800

バッテリーパックは消耗品につき保証対象外とさせていただきます。

MAC

注文No.	商品	価格
U9362	Performa 5320(PPC603e(100)/16MB/1.2G/CD4/15CRT(832X624)/外装焼け)	¥14,800
U9246	PowerMac 7200/90(PPC601(90)/32MB/500HD/CD4/外装焼け)…………	¥8,800
U9281	PowerMac 7200/120(PPC601(120)/32MB/1.2GHD/CD4/外装焼け・キズ)‥	¥10,800
U9351	PowerMac 7300/166(PPC604e(160)/16MB/2GHD/CD12/外装キズ/外装焼け(全体)/マウス無)	¥12,800
U9482	PowerMac 7300/180(PPC604e(180)/32MB/2GHD/CD12/マウス無し/外装焼け/擦り傷)	¥21,800
U9396	PowerMac 7600/120(PPC604(120)/49MB/1.2GHD/CD4/外装焼け)………	¥18,000
U9333	PowerMac 7600/132(PPC604(132)/16MB/1.2GHD/CD8/外装焼け)………	¥10,800
U9342	PowerMac 7600/200(PPC604e(200)/32MB/2GHD/CD12/外装焼け/擦り傷)	¥18,000
U9443	PowerMac 8100/100(PPC601(100)/32MB/700HD/CD2/外装焼け/外装キズ)	¥19,800
U9274	PowerMac 8100/100AV(PPC601(100)/32MB/1GHD/CD2/外装焼け/外装キズ)	¥12,800
U2054	PowerMac 8500/120(PPC604(120)/32MB/2GHD/CD/汚れ/外装焼け/マウス無し)	¥23,800
U9244	PowerMac 8500/120(PPC604(120)/80MB/2GHD/CD4)……………	¥25,000
U9458	PowerMac 8500/132(PPC604(132)/80MB/1GHD/CD4)……………	¥19,800
U9404	PowerMac 8500/180(PPC604e(180)/64MB/2GHD/CD8)…………	¥32,800
U9358	PowerMac 8600/200(PPC604e(200)/32MB/2GHD/CD/ZIP付(動作保証無し)/外装焼け)	¥46,800
U9392	PowerMac 9500/132(PPC604(132)/64MB/2GHD/CD4/外装焼け)……	¥37,800
U9301	PowerMac 9500/150(PPC604(150)/32MB/2GHD/CD4/外装焼け)……	¥41,800
U9367	WorkgroupServer8550/200(PPC604(200)/96MB/2GHD/CD8/外装焼け)・	¥52,800
U1367	PowerMac G3/DT266(PPCG3(266)/32MB/4GHD/CD24/ZIP付き/外装キズ)	¥49,800
U9420	PowerMac G3/MT266(PPCG3(266)/160MB/6GHD/CD24/ZIP/外装焼け)	¥69,800
U9483	PowerMac G3/MT300(PPCG3(300)/128MB/4GHD/CD24/タワー型)……	¥89,800
U9363	PowerMac 4400/200(PPC603e(200)/32MB/2GHD/CD12/リアカバー無し/外装焼け/マウス無)	¥22,800

キーボードは別売となります。

PowerBook

注文No.	商品	価格
U3562	PowerBook 190C(68LC040(66)/8MB/500HD/10.4DSTN)…………	¥14,800
U3776	PowerBook 1400CS/117(PPC603e(117)/48MB/1GHD/CD6/SVGA/外装キズ/11.3FSTNカラー)	¥38,800
U6133	PowerBook 1400CS/117(PPC603e(117)/16MB/770HD/CD6/SVGA/11.3DSTNカラー)	¥34,800
U6470	PowerBook 5300CS(PPC603e(100)/64MB/770HD/VGA/10.4パッシブカラーマトリスク)	¥25,000
U9430	PowerBook 5300C/100(PPC603e(100)/16MB/750HD/VGA/リアカバー無し/10.4CTFTカラー)	¥23,800
U3066	PowerBook 5300C(PPC603e(100)/32MB/750HD/10.4CTFTカラー)……	¥19,800

ディスプレイ

注文No.	商品	価格
U4769	F14T2H(NEC)(14.1TFT:1024X768max/外装焼け/3列15pin)…………	¥34,800
SH445	PC-KM174R(NEC)(17マルチ:1280X1024max/0.25mm/外装キズ/外装焼け/3列15pin/5BNC)	¥13,800
U4826	TFT450(COMPAQ)(14.5TFT液晶:1024X768max/外装焼け/3列15pin)‥	¥32,800
U4561	D1025TM(DELL)(17マルチ:1280X1024max/外装キズ/外装焼け/3列15pin)	¥9,800
U4446	D-1726T-HS(トリニトロン)(DELL)(17マルチ:1024X768max/3列15pin)…	¥11,800
U4376	FLEXSCAN 52F(フラット)(NANAO)(15マルチ:1280X1024max/0.26mm /外装焼け/3列15pin/5BNC)	¥7,800
U4222	FLEXSCAN 56TS(トリニトロン)(NANAO)(17マルチ:1600X1200max/0.25mm/3列15pin/5BNC)	¥15,800
U4274	FLEXSCAN E75F(フラット)(NANAO)(21マルチ:1600X1200max/0.28mm/外装焼け/3列15pin/5BNC)	¥31,800
U4823	EV700(GATEWAY)(17マルチ:1280X1024max/外装キズ/3列15pin)……	¥12,800
SH534	CPD-220VS(SONY)(17マルチ:1280X1024max/3列15pin)…………	¥12,800
U4512	GDM-20SE2T(SONY)(20マルチ:1600X1200max/3列15pin/5BNC)……	¥29,800
U4333	GDM-2038(SONY)(21マルチ:1280X1024max/3列15pin/5BNC)………	¥37,800
U4517	FMV-DP846(富士通)(15マルチ:1024X768max/外装焼け/3列15pin)……	¥7,300
SH402	RD-17M II(三菱)(17マルチ:1280X1024max/0.27mm/3列15pin)………	¥16,000
U4529	RD-17M(三菱)(17マルチ:1280X1024max/0.28mm/3列15pin)………	¥14,000
U4616	RD-17GX II(ダイヤモンド)(三菱)(17マルチ:1280X1024max/0.25mm/外装焼け/3列15pin)	¥13,800
U4519	RD-21G II(ダイヤモンド)(三菱)(21マルチ:1600X1200max/0.28mm/3列15pin/5BNC)	¥30,000
U4146	MT-8617E(飯山電機)(17マルチ:1600X1200max/0.26mm/外装焼け/3列15pin/5BNC)	¥12,800
U4742	RST13X(AKIA)(13.3TFT液晶:1024X768max/ヤケ/3列15pin)…………	¥29,800
U4487	CPD-15F23(GATEWAY)(15マルチ:1024X768max/3列15pin)…………	¥7,300
U4592	D2845(HP)(21マルチ:1280X1024max/外装キズ/3列15pin)…………	¥28,800

プリンタ

注文No.	商品	価格
U5135	LBP-740(キヤノン)(レーザー:A3対応)…………………………	¥24,800
P0142	LBP-750(キヤノン)(レーザー:A3対応/付属品付)………………	¥25,800
P0102	LBP-A309G2(キヤノン)(レーザー:A3対応)……………………	¥23,800
U5152	LBP-2030(キヤノン)(レーザー)………………………………	¥48,000

トナー等は消耗品につき標準添付されておりません。別途、ご確認下さい。

商品の状態により価格の変動がある事がございます。
詳しくはTELかネット上でご確認ください。

万一、品切れの際はご容赦ください。(9月27日現在)

こちらの在庫リストは通販専用となります。店舗在庫は各店舗へご来店の上ご確認ください。

秋葉原駅前店

千代田区外神田1-15-6 ミツウロコビル4F



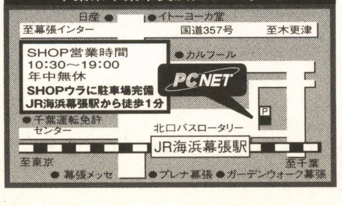
大阪日本橋店

浪速区難波中2-1-7 なんば東ビル
TEL.06-4396-1401



幕張店

千葉県千葉市美浜区ひび野1-9



パシフィック ネット

株式会社パシフィックネット
本社 〒144-0051 東京都大田区西蒲田7-10-1
相互西蒲田ビル TEL.03-3733-1441 FAX.03-3733-1440
通販営業時間／月～金 9:30～19:00 土・日・祝日 11:00～19:30

計測機器を各ショップのみで販売中。
詳しくは各ショップまでお問い合わせください。

iモードでショップ最新情報検索
http://www.prins.co.jp/i/

Q10　携帯・PHSを持っていますか？
①両方持っている ②携帯 ③PHS ④購入予定 ⑤予定なし

Q11　Q10で④の方のみ，どちらを購入する予定ですか？
①携帯 ②PHS ③両方 ④未定

Q12　使用中のプログラミング言語は？
※枠内に一つご記入ください

Q13　これから勉強したいプログラミング言語は？
※枠内に一つご記入ください

Q14　今月号の本誌で面白かった記事を三つ，下の表からお答えください

Q15　今月号の本誌でつまらなかった記事を三つ，下の表からお答えください

ご協力ありがとうございました。

■今月号の記事一覧

① 編集部の学校訪問記
② AIBOを無線でコントロール！
③ ロボ道場番外編「ロボカップトイズ」
④ マンガで学ぼう！パソコン レクチャー・ザ・ビギニング
⑤ Technical News Spot
⑥ PROGRAMMING Get＆Run
⑦ はじめようMicrosoft ExcelでのBASICプログラミング
⑧ IE＋DirectAnimationマルチメディア・プログラミング
⑨ 図解！Visual Basicゲーム作成術
⑩ 特選投稿プログラム・コーナー
⑪ CMM試聴室
⑫ 明日のスタープログラマー＆Programmer’s Room
⑬ Dr.Dのベストプログラム研究室
⑭ PS2用「BASIC STUDIO パワフルゲーム工房」入門
⑮ Catlikeのもっと身近にプログラミング！
⑯ キミにもできる！「iアプリ」制作
⑰ 松原拓也の自作一代記
⑱ テクノ・トイズ・ラボラトリ
⑲ 神奈川のりこのホームページとかに夢中
⑳ レゴ道場　チャレンジ！レゴ・マインドストーム
㉑ よいこのLinux修行
㉒ DTM Art Studio　パソコンで音楽をたのしもう！
㉓ あくせすROOM
㉔ SOFTWARE MAKER HOT INFORMATION
㉕ 表紙
㉖ OVER FLOW

キリトリ線

マイコンBASIC Magazine　プログラム投稿応募カード

言語名（音源名）					
プログラム名（曲名）					
氏　名		ペンネーム		年齢	歳
住　所	〒				
TEL	※連絡可能な時間帯			応募年月日	年　月　日

キリトリ線

郵便はがき

50円切手をお貼り下さい

141-8715

（受取人）

東京都品川区東五反田1−11−15

電波新聞社

月刊 マイコンBASICマガジン 編集部

11月号 行

（フリガナ）		年齢	性別
ご氏名		歳	男・女
□□□-□□□□		電話	
ご住所	都・道 府・県		

職業：小学生／中学生／高校生／高専生／短大生／大学生／専門学校生／予備校生／アルバイト／公務員／会社員／その他（　　　）

本誌を 毎月・時々・初めて 読む　何人で読みますか（　）人

自分のホームページを　持っている・持つ予定・持っていない・予定なし

本誌のほかによく読む雑誌は？［　　　］

ベーマガお楽しみプレゼント⑪応募欄（86ページ欄外）	希望番号	品名
	番	

キリトリ線

月刊マイコンBASICマガジン⑪ 愛読者カード

★OFコーナー宛 ♥あくせすROOM宛 ■質問・感想（本誌への掲載／可・不可）

ペンネーム：

■アンケート回答記入欄■

※黒の筆記用具でわかりやすく，はっきりとご記入ください

Q1	□	Q2	□	［メーカー名 ］				
Q3	□	Q4	□	Q5	□			
Q6	□	□	□	□	□	□	□	□
Q7	□	□	□	□	□	［商品名 ］		
Q8	□	□	□	□	□	□	□	□
Q9	□	□	□	□	□	□	□	□
Q10	□	Q11	□					
Q12	［ ］	Q13	［ ］					
Q14	□	□	□	Q15	□	□	□	

キリトリ線

■アンケートにご協力ください。

※左の愛読者カード・記入欄に，以下の質問に該当する番号をご記入ください。複数回答の質問で，回答欄をすべて埋める必要はありません。

Q1　メインで使っているパソコンは？
①デスクトップ ②ノート ③持っていない

Q2　パソコンの種類は？（Q1で③以外の方）
①自作 ②ショップブランド ③メーカー製
※枠内にメーカー名も記入してください

Q3　メインで使っているOSは？（Q1で③以外の方）
①Windows 95 ②Windows 98 ③Windows Me ④Windows 2000 ⑤Windows NT ⑥Macintosh ⑦UNIX ⑧Linux ⑨その他

Q4　パソコン歴は？
①1年未満 ②1～2年 ③3～5年 ④6～10年 ⑤11年以上 ⑥なし

Q5　インターネット歴は？
①1年未満 ②1～2年 ③3～5年 ④6～10年 ⑤11年以上 ⑥なし

Q6　パソコンの主な使用目的は？（複数回答可）
①プログラミング ②ゲーム ③インターネット ④メール ⑤ミュージック ⑥学習 ⑦CG ⑧その他

Q7　これから買いたいと思うコンピュータ機器は？（複数回答可）
①パソコン本体 ②プリンタ ③スキャナ ④デジタルカメラ ⑤その他
※わかれば，商品名も記入してください

Q8　持っているゲーム機は？（複数回答可）
①プレイステーション（PSone含） ②プレイステーション2 ③ドリームキャスト ④ニンテンドウ64 ⑤ゲームボーイ ⑥ゲームボーイカラー ⑦ゲームボーイアドバンス ⑧ワンダースワン ⑨ワンダースワンカラー ⑩その他

Q9　これから買いたいと思うゲーム機は？（複数回答可）
①プレイステーション（PSone含） ②プレイステーション2 ③ドリームキャスト ④ニンテンドウ64 ⑤ゲームキューブ ⑥Xbox ⑦ゲームボーイカラー ⑧ワンダースワンカラー ⑨ゲームボーイアドバンス

※裏面へつづく→

■メモ欄…プログラム担当者・編集部へのご意見・ご感想など

アクションRPGのファルコムから！

イース、ザナドゥ、ブランディッシュ・・・etc。数々の名作を輩出してきたファルコムのアクションRPG史に、今、新たなる歴史が刻まれる――。

完全新作「ツヴァイ!!」

12月発売！

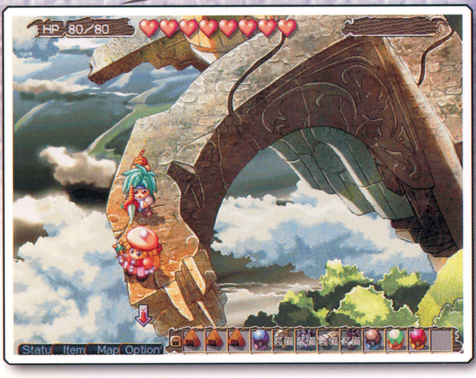
WindowsMe/98/95
(Windows2000対応)



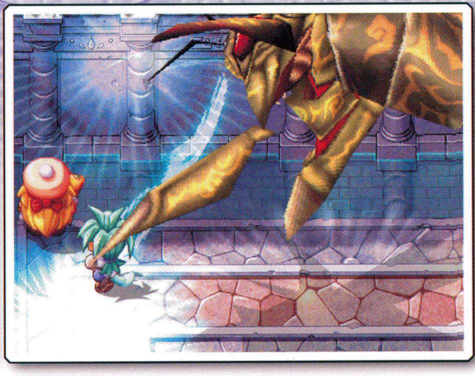
さあ！2人で大冒険!!

豊かな自然と遺跡に彩られた浮遊大陸「アルジェス」。
永きに渡る平穏が揺らぎ始めた時、2人の大冒険の幕が明ける―。

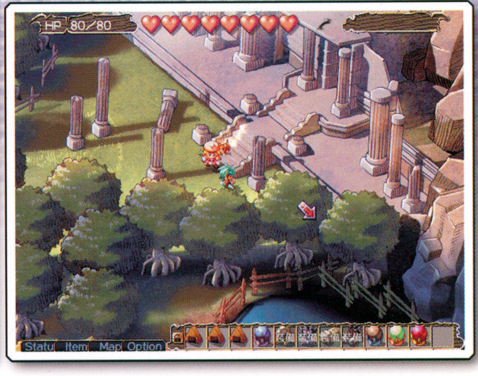
連綿と受け継がれるスピリット！パソコンゲームにアクションRPGというジャンルを切り開いたファルコムだからこそできる、熟練と革新が生み出す新ステージ「ツヴァイ!!」に乞うご期待！



▲2人が冒険する舞台は山あり谷あり、空中あり!? さらには火の中、水の中！隅々まで歩き回って、普通にプレイしたら見ることができない隠されたイベントを探し出そう！



▲ボス戦には定評があるファルコムのアクションRPG！「ツヴァイ!!」には、画面に収まらないほどのファルコムゲーム史上最大クラスのボスが続々！



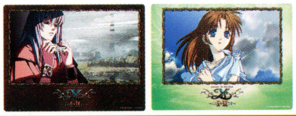
▲村の外れに建立されている謎の神殿。外の世界だけでなく、村の中にも謎がいっぱい。隠された秘宝を見つけるのも、冒険の目的の1つ！

「ツヴァイ!!」の最新情報は、ファルコムホームページをご覧ください。 **http://www.falcom.com/**

PCゲーム定番

PCゲーム定番キャンペーン(9/29〜11/30)

キャンペーン期間中に、イースⅠ・Ⅱ完全版他、ファルコム製品をキャンペーン実施店で購入された方に、薄くて軽い超便利な最新型マウスシートを全員プレゼント！
詳しくは店頭に貼られてあるポスターをご覧ください。



キャンペーン実施店は、ファルコムホームページのショップリストに掲載中！お近くの方は要チェック！

キャンペーン参加店様募集中!
お気軽にお問い合わせください。

Falcom®
NIHON FALCOM CORPORATION
〒190-0012 東京都立川市曙町1-14-13
☎042(527)6501 http://www.falcom.co.jp/



定価550円 本体524円 雑誌18361-11 T1118361110555

第二十巻第十一号通巻二三三号 昭和五十七年十二月六日第三種郵便物認可 二〇〇一年十月二十五日印刷 二〇〇一年十一月一日発行

電波新聞社 〒141-8715 東京都品川区東五反田一-十一-十五 電話〇三(三四四五)六一一一

2001

わかる!動かせる!
プログラムが組める雑誌

マイコンBASIC Magazine

11

AIBOを無線でコントロール!

http://www.basicmagazine.net/

電波新聞社